信息科学技术学术著作丛书

大数据搜索与挖掘

张华平　高　凯
黄河燕　赵燕平　著

科学出版社

(TP-6591.01)

www.sciencep.com

ISBN 978-7-03-040318-6

9 787030 403186 >

定 价：90.00 元

工程技术分社
电 话：010-64023167
E-mail: weiyingjie@mail.sciencep.com

销售分类建议：计算机、人工智能、大数据搜索

信息科学技术学术著作丛书

大数据搜索与挖掘

张华平　高　凯　黄河燕　赵燕平　著

科学出版社

北　京

内 容 简 介

本书以作者团队十余年在大数据搜索与挖掘领域所作的研究与应用工作为内容，介绍大数据搜索与挖掘的研发成果，内容涵盖大数据处理概论、中文自然语言处理、网络信息预处理、网络情报挖掘(包括网络语言分析、新特征语言抽取、汉语词法分析、文本自动分类、自动聚类、自动摘要、关键词抽取)、网络精准搜索(信息检索模型、句子搜索)、人物搜索等方面的研究成果。从大数据这座金矿中挖掘有价值的信息，是本书的目的所在。全书体系完整，内容新颖，条理清晰，组织合理，理论与实践并重，突出原创的研究成果与实际应用。

本书可为高校计算机专业、计算机语言学专业和人工智能专业等师生的教学和科研工作提供帮助，也可为从事大数据搜索与挖掘、中文自然语言处理、信息检索与搜索引擎技术研发的工程技术人员和希望了解上述技术的爱好者等提供参考。

图书在版编目(CIP)数据

大数据搜索与挖掘/张华平等著. —北京：科学出版社，2014

(信息科学技术学术著作丛书)

ISBN 978-7-03-040318-6

Ⅰ. 大… Ⅱ. 张… Ⅲ. ①情报检索-研究②数据采集-研究

Ⅳ. ①G354②TP274

中国版本图书馆 CIP 数据核字(2014)第 060074 号

责任编辑：魏英杰 / 责任校对：彭 涛

责任印制：张 倩 / 封面设计：陈 敬

科 学 出 版 社 出版

北京东黄城根北街 16 号

邮政编码：100717

http://www.sciencep.com

新科印刷有限公司 印刷

科学出版社发行 各地新华书店经销

*

2014 年 5 月第 一 版 开本：720×1000 1/16

2014 年 5 月第一次印刷 印张：19 1/2

字数：391 000

定价：90.00 元

(如有印装质量问题，我社负责调换)

作 者 简 介



张华平 1978年出生。工学博士，北京理工大学副教授。毕业于中国科学院计算技术研究所。汉语词法分析系统ICTCLAS创始人，ICTCLAS在国家973评测和第一届国际汉语分词大赛中综合得分均获得第1名。

主要从事大数据搜索与挖掘、自然语言处理、信息检索等方面的研究工作，主持或参与国家自然科学基金、863、973、242等十余项课题。曾先后获得2010年度钱伟长中文信息处理科学技术奖一等奖，中国科学院院长优秀奖、中国科学院计算技术研究所所长特别奖，是中国科学院计算技术研究所“百星计划”首批入选者。



高凯 1968年出生。工学博士。毕业于上海交通大学计算机应用技术专业，河北省重点学科“计算机软件与理论”中“信息检索与云计算”方向学术带头人。

主要从事大数据搜索与挖掘、自然语言处理、网络信息检索、社会网络计算等领域的研究工作。



黄河燕　1963 年出生。工学博士，教授、博士生导师，现任北京理工大学计算机学院院长、国家高技术研究发展计划(863 计划)主题专家组成员、教育部计算机专业指导委员会委员、中国人工智能学会副理事长、中国中文信息学会副理事长兼自然语言处理专业委员会主任。

主要从事自然语言处理和机器翻译、智能处理系统等领域的研究，承担了近 20 项国家级科研攻关项目和大型工程应用，以及国际合作项目，获得国家科学技术进步奖一等奖、国家经济贸易委员会九五技术创新优秀项目奖、中央国家机关十大杰出青年等荣誉和奖励。



赵燕平　1956 年出生。北京理工大学教授，国家人力资源和社会保障部职业技能鉴定中心电子商务专业委员会专家，中国电子学会健康物联专委会专家。北京理工大学大数据搜索与挖掘实验室副主任，曾任联合国开发计划署(UNDP)“中国可持续发展网络计划”项目专家。主持参与了多个科研和工程项目。

《信息科学技术学术著作丛书》序

21世纪是信息科学技术发生深刻变革的时代，一场以网络科学、高性能计算和仿真、智能科学、计算思维为特征的信息科学革命正在兴起。信息科学技术正在逐步融入各个应用领域并与生物、纳米、认知等交织在一起，悄然改变着我们的生活方式。信息科学技术已经成为人类社会进步过程中发展最快、交叉渗透性最强、应用面最广的关键技术。

如何进一步推动我国信息科学技术的研究与发展；如何将信息技术发展的新理论、新方法与研究成果转化为社会发展的新动力；如何抓住信息技术深刻发展变革的机遇，提升我国自主创新和可持续发展的能力？这些问题的解答都离不开我国科技工作者和工程技术人员的求索和艰辛付出。为这些科技工作者和工程技术人员提供一个良好的出版环境和平台，将这些科技成就迅速转化为智力成果，将对我国信息科学技术的发展起到重要的推动作用。

《信息科学技术学术著作丛书》是科学出版社在广泛征求专家意见的基础上，经过长期考察、反复论证之后组织出版的。这套丛书旨在传播网络科学和未来网络技术，微电子、光电子和量子信息技术、超级计算机、软件和信息存储技术，数据知识化和基于知识处理的未来信息服务业，低成本信息化和用信息技术提升传统产业，智能与认知科学、生物信息学、社会信息学等前沿交叉科学，信息科学基础理论，信息安全等几个未来信息科学技术重点发展领域的优秀科研成果。丛书力争起点高、内容新、导向性强，具有一定的原创性；体现出科学出版社“高层次、高质量、高水平”的特色和“严肃、严密、严格”的优良作风。

希望这套丛书的出版，能为我国信息科学技术的发展、创新和突破带来一些启迪和帮助。同时，欢迎广大读者提出好的建议，以促进和完善丛书的出版工作。

中国工程院院士
原中国科学院计算技术研究所所长
李国杰

序

随着以微博、Twitter 与 Facebook 等为代表的新型社交网络的迅猛发展，大数据已经成为新一轮科技革命中信息技术发展的新趋势，将对国家经济与社会产生重大的影响。大数据分为结构化大数据与非结构化大数据，结构化大数据在 20 世纪末期已经有较好的研究，产生了关系型数据库以及数据仓库等相对成熟的技术与产品，也造就了 IBM、Oracle 等跨国大企业。以文本、多媒体信息为主要元素的非结构化大数据，体量是结构化大数据的十倍甚至更大，但非结构化大数据的分析处理挑战更大。非结构化大数据的生产、搜索、挖掘与分析已经成为了当前研究的热点与难点。

《大数据搜索与挖掘》一书顺应了非结构化文本大数据搜索与挖掘的内在要求，作者结合十余年的科学研究与工程技术开发的积累，有针对性地分享了他们的研究成果与工程经验。该书是目前有关文本大数据搜索与挖掘领域比较全面和系统的著作，也是诸多大数据挖掘书籍中为数不多的穿插大量实践应用案例和场景的著作。该书全方位整理、总结、分享了学术思想与项目经验，并提供了相关程序代码和技术支持；通过案例分享，更好地为相关专业的学习者与研究者提供帮助。有理由相信，该书能对有志于从事大数据搜索与挖掘的学者与工程师提供理论和实践支撑。

张华平博士自主研发的 ICTCLAS 是目前最受欢迎、应用最广泛的中文词法分析系统，已经成为中文信息处理领域的标杆。张华平博士是国内大数据搜索与挖掘方面年轻活跃的学者，在学术界与产业界有着重要的影响，包括中央电视台在内的媒体均进行过报道。在这部著作中，张华平所在团队结合他们在自然语言处理、大数据搜索与挖掘、数据运营实践中积累的大量项目经验，围绕大数据文本挖掘中的思路、方法、技巧与应用，提出了知著、显微、晓意的大数据解决之道，与我多年在信息处理尤其是少数民族语言的信息处理方面有着异曲同工之妙。

这是一本难得一见的理论联系实际的学术著作，必将对国内大数据搜索与挖掘的研究和发展起到积极的推动作用。

中国工程院院士

吾守尔·斯拉木

前　言

随着互联网的迅猛发展和信息传播手段的丰富，人类已经进入一个信息爆炸的大数据时代。面对日益增长的网络大数据，高效快捷地获取有用信息，已经成为时代发展的迫切需要。本书以作者及其研发团队十余年来在大数据搜索与挖掘领域所作的研究与应用工作为主线，介绍大数据搜索与挖掘的基本原理，就网络信息智能处理技术中的 Web 数据挖掘、自然语言处理（如分词、词典管理、主题词抽取、摘要、分类、聚类、有意义串挖掘等）技术、信息检索与知识发现等进行阐述，并着重从中文自然语言处理的角度来提高信息检索与挖掘的质量，力争开发出性能优异的大数据搜索与挖掘系统。

全书突出原创性的研究成果，理论与实践并重，强调算法、技术实现与实际应用，其主要内容涉及作者团队近年来的研究成果，囊括了大数据搜索、中文自然语言处理、网络信息预处理、网络情报挖掘（包括网络语言分析、新特征语言抽取、汉语词法分析、文本自动分类、自动聚类、自动摘要、关键词抽取）、网络精准搜索（包括信息检索模型、句子搜索）等方面的研究成果，并从实际应用出发，介绍作者参与研发的科研成果及在相关部门的应用。

全书成果主要涉及张华平及其指导的研究生的科研成果，有些章节内容直接来自成果论文；高凯完成了有关算法、网络信息获取、搜索引擎等内容以及部分相关工作综述、扩展阅读等的撰写工作；黄河燕教授审阅了全书；赵燕平教授完成了最后的统稿工作。在这里，对导师以及相关工作的合作者白硕研究员、刘群教授、程学旗教授等表示衷心感谢。在本书的写作与相关科研课题的研究工作中，得到了多方面的支持与帮助，并参考了作者指导的部分研究生（贺敏、张京阳、王思力、黄玉兰、秦鹏、刘志华等）以及课题组成员于满泉等的博士学位论文和赵燕平老师一些学生的硕士学位论文，而有关信息检索模型部分系摘自由张华平等翻译、Grossman 和 Frieder 合著的信息检索专著 *Information Retrieval: Algorithms and Heuristics*（*Second Edition*）中的部分章节。另外，国内外众多的信息检索与数据挖掘方面的研究成果和相关网站亦为本书提供了帮助，本书的顺利完成也得益于参阅了大量的相关工作及研究成果，在此谨向这些文献的作者以及为本书提供帮助的人致以诚挚的谢意和崇高的敬意。

本书得到 2013 年国家自然科学基金（编号：61272362）、国家 973 重点基础研

究发展计划(编号:2013CB329606)、2012 年河北省科技支撑计划(编号:12213516D)、2012 年新疆维吾尔自治区高新技术计划(编号:201212124)、2013 年河北省自然科学基金(编号:F2013208105)的资助。在本书写作过程中,也得到了科学出版社的大力支持和帮助,在此一并表示衷心感谢。

作为本书成果的配套网站和相关资料的下载基地,课题组的官方网站(http://www.nlpir.org)提供实际成果演示与各类资源的下载。

限于作者的学识水平,书中不妥之处在所难免,恳请广大读者批评指正。

作 者

2014 年 2 月

目　　录

《信息科学技术学术著作丛书》序

序

前言

第 1 章　绪论 …… 1

1.1　大数据 …… 4

1.2　云计算及 Hadoop 简介 …… 5

1.3　Web 搜索、全文索引与 Lucene 简介 …… 6

1.3.1　Web 搜索 …… 7

1.3.2　全文索引 …… 9

1.3.3　Lucene 简介 …… 11

1.4　大数据挖掘 …… 13

1.5　本书主要内容及其知识点 …… 14

1.6　本章小结 …… 16

参考文献 …… 17

第 2 章　大数据搜索挖掘综述 …… 18

2.1　常用的信息检索模型 …… 18

2.1.1　传统布尔检索与扩展布尔检索模型 …… 18

2.1.2　向量空间模型 …… 19

2.1.3　概率检索模型 …… 23

2.1.4　语言模型 …… 24

2.2　自然语言理解与处理概述 …… 26

2.3　中文词法分析中的分词处理 …… 28

2.3.1　基于词典和规则的汉字分词 …… 30

2.3.2　基于大规模语料库的统计学习的分词方法 …… 30

2.3.3　规则和统计方法相结合的汉字分词方法 …… 32

2.4　未登录词及其识别 …… 32

2.4.1　命名实体及其识别 …… 33

2.4.2　未登录词与新词识别 …… 34

2.5 有意义串及其识别 …… 36
2.6 词典组织与管理 …… 37
2.6.1 基于 Trie 索引树的词典管理 …… 37
2.6.2 基于哈希表的词典管理 …… 38
2.7 文本分类 …… 39
2.8 文本聚类 …… 41
2.8.1 文本表示 …… 41
2.8.2 相似度度量 …… 42
2.8.3 聚类算法体系 …… 43
2.9 话题识别与跟踪 …… 46
2.10 句子及其检索 …… 48
2.10.1 传统的文档检索方法 …… 48
2.10.2 信息过滤方法 …… 48
2.10.3 分类方法 …… 49
2.10.4 语义比较方法 …… 49
2.10.5 隐马尔可夫模型方法 …… 50
2.10.6 自动文摘方法 …… 51
2.11 句子级新信息检测 …… 52
2.11.1 词重叠度 …… 52
2.11.2 最大区间相关度 …… 52
2.11.3 余弦冗余度 …… 52
2.11.4 命名实体触发方法 …… 53
2.11.5 统计机器翻译模型 …… 53
2.11.6 LexRank 方法 …… 54
2.12 本章小结 …… 55
参考文献 …… 56
第 3 章 大数据检索与分词 …… 62
3.1 概述 …… 62
3.2 分词对中文信息检索的影响 …… 63
3.3 分词精度与检索性能的关系 …… 66
3.4 大数据应用环境下中文信息检索的分词算法及其特点 …… 70
3.4.1 分词算法的时间性能要求高 …… 70

3.4.2　分词正确率的提高并不一定带来检索性能的提高 …… 70
3.4.3　分词切分粒度需在查询扩展层面进行相关处理 …… 70
3.4.4　未登录词识别的准确率要比召回率更重要 …… 71
3.5　基于双数组 Trie 树优化算法的词典 …… 72
3.5.1　双数组 Trie 树算法介绍及其优化 …… 72
3.5.2　利用优化的双数组 Trie 树算法组织词典 …… 74
3.5.3　实验结果与分析 …… 76
3.6　本章小结 …… 77
参考文献 …… 78
第 4 章　基于层次隐马尔可夫模型的浅层词法分析 …… 80
4.1　概述 …… 80
4.2　英文浅层分析的实现 …… 81
4.2.1　英文断句与词汇切分 …… 81
4.2.2　词性标注 …… 82
4.2.3　词干抽取与词形还原 …… 83
4.3　停用词处理与特征词选择 …… 84
4.3.1　停用词处理 …… 84
4.3.2　特征词选择 …… 85
4.4　基于层次隐马尔可夫模型的汉语浅层分析及其应用 …… 86
4.4.1　层次隐马尔可夫模型 …… 87
4.4.2　基于类的隐马尔可夫分词算法 …… 89
4.4.3　N 最短路径的切分排歧策略 …… 90
4.4.4　未登录词的隐马尔可夫识别方法 …… 91
4.5　汉语词法分析系统 ICTCLAS 性能实验与分析 …… 93
4.5.1　词法分析与层次隐马尔可夫模型 …… 94
4.5.2　ICTCLAS 在 973 评测中的测试结果 …… 95
4.5.3　第一届国际分词大赛的评测结果 …… 95
4.6　基于单字位置成词概率识别未登录词的算法 …… 96
4.6.1　字的位置成词概率 …… 96
4.6.2　局部二元串频统计 …… 98
4.6.3　有关未登录词识别的实验结果 …… 99
4.7　本章小结 …… 100

参考文献 …… 102
第 5 章 大数据语言新特征发现 …… 104
5.1 概述 …… 104
5.2 基于上下文邻接分析和语言模型的有意义串提取 …… 106
5.2.1 上下文邻接分析 …… 107
5.2.2 语言模型分析 …… 109
5.2.3 重复串发现及处理流程 …… 111
5.2.4 实验设计及结果分析 …… 115
5.3 基于局部性原理的低频有意义串提取 …… 120
5.3.1 有意义串的局部性 …… 121
5.3.2 局部性度量 …… 122
5.3.3 算法流程 …… 123
5.3.4 实验结果与分析 …… 124
5.4 基于伪相关反馈模型的有意义串提取 …… 127
5.4.1 算法的基本思想 …… 128
5.4.2 相关度的定义 …… 129
5.4.3 位置成词概率 PWP 的更新 …… 129
5.4.4 算法流程 …… 129
5.4.5 实验结果及分析 …… 130
5.5 本章小结 …… 133
参考文献 …… 135
第 6 章 大数据聚类与分类 …… 138
6.1 概述 …… 138
6.2 基于关键词提取的搜索结果聚类 …… 139
6.2.1 相关术语简介 …… 139
6.2.2 关键词提取 …… 139
6.2.3 基于关键词的检索结果聚类方法 …… 141
6.2.4 实验结果及分析 …… 142
6.3 基于 K-means 算法的有意义串主题聚类算法 …… 144
6.4 基于邻接串种类的有意义串语境聚类 …… 146
6.5 有意义串对分类的改进 …… 149
6.6 本章小结 …… 153

参考文献 …… 154
第 7 章　大数据文本自动摘要 …… 156
7.1　概述 …… 156
7.2　相关工作综述 …… 156
7.2.1　基于抽取的自动文摘 …… 158
7.2.2　基于理解的自动文摘 …… 160
7.3　基于关键词提取的自动摘要 …… 160
7.3.1　文本预处理 …… 160
7.3.2　停用词处理 …… 161
7.3.3　双数组 Trie 树 …… 162
7.3.4　关键词提取 …… 164
7.3.5　句子切分 …… 166
7.3.6　句子相似度计算 …… 166
7.4　面向主题的自动摘要 …… 167
7.4.1　改进的最大边缘相关度方法 …… 167
7.4.2　面向主题的词特征统计 …… 168
7.4.3　领域主题词表 …… 169
7.4.4　句子间的包含关系 …… 170
7.5　实验与分析 …… 171
7.5.1　稳定性测试 …… 171
7.5.2　时间性能 …… 171
7.5.3　文摘质量 …… 174
7.6　自动摘要应用场景分析及大数据搜索与挖掘软件应用示例 …… 174
7.7　本章小结 …… 176
参考文献 …… 176
第 8 章　JZSearch 大数据精准搜索引擎 …… 178
8.1　概述 …… 178
8.2　JZSearch 大数据搜索引擎系统架构 …… 178
8.3　JZSearch 索引关键技术 …… 180
8.3.1　索引字段类型 …… 180
8.3.2　索引词项的设计 …… 181
8.3.3　索引压缩技术 …… 181

8.3.4 内存交换 …… 184
8.3.5 增量索引 …… 184
8.3.6 数据库检索 …… 185
8.4 JZSearch 搜索技术 …… 187
8.4.1 JZSearch 排序算法 …… 187
8.4.2 JZSearch 结果格式 …… 188
8.4.3 JZSearch 检索语法说明 …… 188
8.5 JZSearch 搜索引擎管理 …… 193
8.5.1 搜索引擎可视化管理客户端 …… 193
8.5.2 客户端管理命令语法 …… 194
8.6 JZSearch 大数据搜索应用案例 …… 194
8.6.1 中国邮政集团名址信息中心首页的邮址垂直搜索 …… 194
8.6.2 河北省标准化研究院的标准搜索 …… 195
8.6.3 中国对外承包工程商会的知识搜索门户 …… 196
8.6.4 富基融通的商品比价搜索 …… 196
8.6.5 微博人物搜索 …… 196
8.6.6 维吾尔语搜索 …… 196
8.7 本章小结 …… 198
参考文献 …… 199
第 9 章 面向大数据的句子检索与新颖性监测 …… 200
9.1 概述 …… 200
9.2 句子检索的查询扩展方法 …… 201
9.2.1 语义扩展 …… 201
9.2.2 伪相关反馈扩展 …… 203
9.2.3 局部共现扩展 …… 204
9.3 语言模型检索 …… 206
9.3.1 概述 …… 206
9.3.2 句子级语言模型及其改进 …… 207
9.4 句子检索实验与分析 …… 207
9.4.1 浅层语言分析的贡献度 …… 207
9.4.2 三种句子检索模型的基准实验 …… 209
9.4.3 查询扩展实验 …… 211

9.5　新信息检测 …… 212
9.5.1　词重叠度及其扩展 …… 213
9.5.2　相似度比较方法 …… 214
9.5.3　信息增强评价方法 …… 215
9.5.4　其他方法 …… 215
9.5.5　新信息检测实验与分析 …… 217
9.6　监督学习条件下的句子检索与新信息检测 …… 219
9.6.1　监督学习环境下的参数调整与阈值设置 …… 219
9.6.2　基于分类的句子检索与新信息检测方法 …… 221
9.6.3　实验与分析 …… 222
9.7　本章小结 …… 224
参考文献 …… 225
第 10 章　人物追踪中的数据预处理与属性抽取 …… 227
10.1　概述 …… 227
10.1.1　研究背景 …… 227
10.1.2　人物追踪及其处理流程 …… 228
10.2　数据预处理 …… 228
10.2.1　数据预处理的主要流程 …… 229
10.2.2　网页正文提取与噪声过滤 …… 229
10.2.3　人名识别 …… 232
10.2.4　人名指代处理 …… 232
10.2.5　人物对应语段的确定 …… 233
10.2.6　时间和时序标签的确定 …… 234
10.3　人物属性抽取 …… 234
10.3.1　人物属性抽取的总体框架 …… 235
10.3.2　标注人物属性抽取语料 …… 235
10.3.3　分类器模型 …… 237
10.4　本章小结 …… 243
参考文献 …… 245
第 11 章　人物模型组织与基于事件的信息处理 …… 246
11.1　概述 …… 246
11.2　人物模型的特征表示 …… 247

11.2.1 属性特征的表示 …… 247
11.2.2 数值特征的表示 …… 247
11.2.3 各项特征的分布规律 …… 249
11.3 人物模型的相似度计算方法 …… 249
11.3.1 基本属性的相似度计算 …… 249
11.3.2 介绍性属性的相似度计算 …… 249
11.3.3 词场的相似度计算 …… 251
11.3.4 人物模型相似度计算 …… 251
11.4 人物模型的同一性判别与合并 …… 252
11.5 实验结果与分析 …… 253
11.5.1 数据集与评测方法 …… 253
11.5.2 实验结果 …… 254
11.6 基于宏观粒度的事件组织 …… 258
11.6.1 宏观粒度事件的特征 …… 258
11.6.2 针对事件特点的话题识别方法 …… 259
11.6.3 基于多层聚类的话题层次化组织方法 …… 261
11.6.4 实验结果与分析 …… 265
11.7 本章小结 …… 270
参考文献 …… 272
附录 A ICTCLAS/NLPIR 2014 汉语分词系统介绍 …… 274
附录 B NLPIR 大数据搜索与挖掘共享开发平台 …… 281

第1章 绪 论

英国科学家詹姆斯·马丁认为，人类的知识在19世纪是每50年增加1倍，20世纪中叶是每10年增加1倍，到20世纪70年代就已经缩短为每5年增加1倍。如今的信息，特别是互联网上的Web信息，更如爆炸般产生，且信息的生产能力已超过了人们对其处理和吸收的能力。正如美国作家奈斯比特在其著作《大趋势》一书中曾指出的那样："我们虽淹没在信息的海洋中，但是却渴求所需的知识。"为什么会出现这种情况呢？主要原因之一是缺乏有效的大数据搜索、挖掘与知识获取手段。因此，帮助人们快速、方便、准确地从大数据海洋中搜索并挖掘到所需信息，已成为时代发展的需要(高凯，2010)。

随着以Twitter与Facebook等为代表的新型社交媒体的出现和迅猛发展，基于互联网的社交媒体已经成为信息传播的重要渠道和载体。截至2012年3月，Twitter共有1.4亿活跃用户，每天约3.4亿条推文。截至2012年5月，Facebook拥有约9亿用户，月活跃用户接近8.5亿。《第31次中国互联网络发展状况调查统计报告》显示：截至2012年12月，我国微博用户规模为3.09亿，网民中的微博用户比例达到54.7%，手机微博用户规模达到2.02亿，新浪微博注册用户已超5亿，用户每日发博量超过1亿条(CNNIC，2013)。社交媒体在短短几年内快速覆盖了全球10亿以上的主流人群，影响着大多数人生活的方方面面，对社会经济生活、管理创新甚至国家战略都产生了深远的影响。互联网已成为人们获取信息的首选途径，也成为大数据的重要来源。

让我们以一分钟为窗口看一组数据，感受一下大数据时代的信息传播。

(1) 每分钟Facebook用户评论数超过50万。每分钟Facebook用户主页状态更新数超过29万，图片上传量接近14万张。按照这个速度计算，只需15分钟，Facebook图片上传量就相当于纽约公共图片档案馆中的图片总量。可见，用户将为Facebook创造数量惊人的内容。

(2) Twitter用户每分钟发布12万条信息。但是，Twitter用户每分钟通过第三方服务上传的图片量并不大，仅为Facebook的1%。

(3) 图片社交网络Flickr每分钟图片上传量为3500张。

(4) 照片分享应用Instagram每分钟图片上传量为1350张。

(5) 地理定位服务FourSquare用户每分钟签到数超过2000人。

(6) 轻博客Tumblr每分钟博文发布量为1000篇，博文转发量为300篇。

(7) 点评网站Yelp每分钟接收20条新评论，每小时接收120张新图片。

(8) 视频网站 YouTube 每分钟上传大约 48 小时的视频。

(9) 2012 年,天猫与淘宝的“双十一”交易总额为 53 亿元,“双十一”全天,支付宝交易笔数超 1 亿笔,最高峰时处理交易数达 20.5 万笔/分钟。

(10) 新浪微博用户平均每天发布超过 1 亿条微博内容,每分钟平均处理微博 7 万条。

何谓大数据? 研究机构 Gartner 给出了这样的定义:大数据是需要新处理模式才能具有更强的决策力、洞察发现力和流程优化能力的海量、高增长率和多样化的信息资产;大数据通常用来形容大量的非结构化和半结构化数据。

大数据的主要特点如下:

(1) 数据量巨大——从 TB 级别,跃升到 ZB 级别;伴随着数据规模的剧增,数据的价值密度在减小,从大数据中挖掘有价值的知识,需要在搜索与挖掘方面采用一些新的技术手段与方法。

(2) 数据多样化——传统的数据挖掘主要是以结构化数据为主,而大数据更加多样化,结构化数据、半结构化以及非结构化数据融合,来源多样化,数据质量良莠不齐。利用成熟的关系型数据仓库与数据挖掘手段,基本解决了结构化数据的存储、检索与挖掘问题,已经成就了 Oracle、SQL Server、IBM DB2 等成熟的产品。目前,结构化数据不过是冰山一角,规模更大的是非结构化数据,其规模是结构化数据的十倍乃至百倍。海量异构文本内容数据(如网页、微博与社交网络、网络日志、视频、图片、RFID、地理位置信息等)的处理挑战更大,还缺乏成熟的技术,需要运用新的处理和发现模式,来挖掘大数据背后隐藏着的知识。因此,大数据的处理更大意义上是对非结构化数据的处理,本书的主旨也是聚焦于非结构化大数据分析处理技术的研究。

(3) 数据速度变更快——大数据时代之前的处理数据一般都是静态化的内容,或者数据的产生与变化速度较低,采用批处理的方式即可解决。而微博等社交网络数据以及电子商务等交易数据是以实时数据流的方式出现的,采用静态的处理架构已经无法应对大数据稍纵即逝的特点。

面对大数据的新特点,如何解决大数据规模大、速度快、多样化以及价值密度低等挑战? 我们的观点是:采用大数据分析方法学,即从客观存在的全量超大规模、多源异构、实时变化的微观数据中,利用自然语言处理、信息检索、机器学习等技术抽取知识,转化为智慧的方法学。下面以图 1.1 为例来研究我们对大数据正确认识的方法论。一般的认识过程包括以下三个有序的过程:

(1) 知著:从宏观上认识大数据的轮廓、结构、宏观特征。首先能从图中发现总体是一个“大”字,而且是由多个一样的“小”字按照“大”字的结构有机组成的;大数据处理的要领就在于开始时忽略细枝末节,从宏观上尽快挖掘出统一的结构特性,获取综合的共同点,发现相关性。

(2) 显微:细究大数据的微观构成特征。从微观可以发现每个细节都是由“小”字组成,但仅仅聚焦于微观的“小”字,会跌入盲人摸象式的片面误区。因为大数据本身的价值密度极低,而且大数据处理的效率要求实时,不可能针对每个微观部分都做细致的分析处理,更多的是分析关键节点。正如我们在做微博分析的时候,会发现大量用户的内容实际上是没有分析价值的,而单一用户有分析价值的内容也是有限的。

(3) 晓意:在大数据中引入语言语义的分析与理解。如果不懂汉语,不懂“大”与“小”的语义,不清楚“大中有小”、“小中喻大”的辩证关系,整个图的精妙所在也就无从知晓。大数据的关键在于非结构化数据,因此将自然语言处理的技术引入大数据处理是大势所趋。

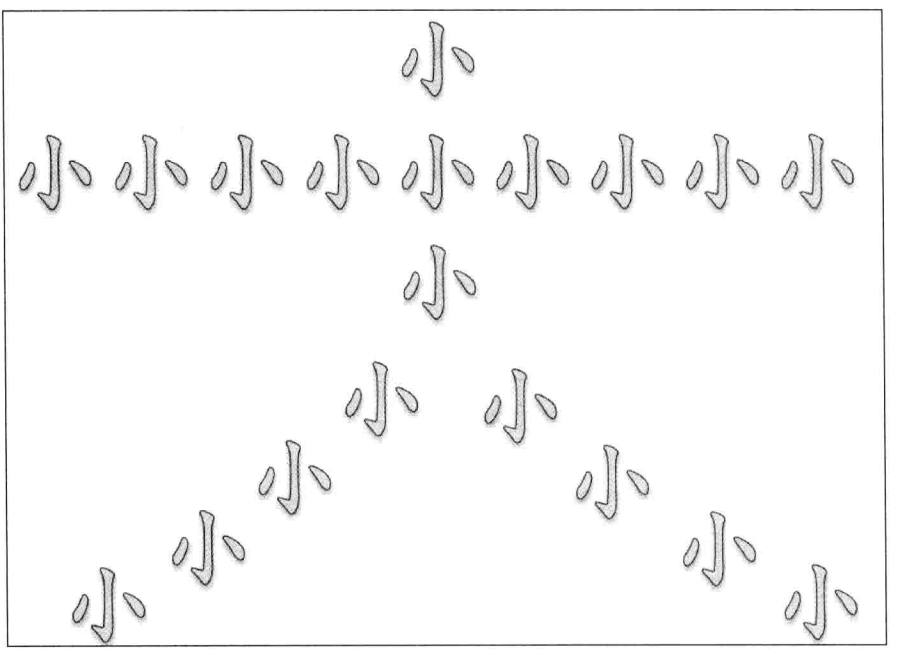


图 1.1 大数据的应对之道

随着大数据时代的来临,大数据也吸引了越来越多的关注。另一方面,近年来,随着语料库语言学的兴起,尤其是随着以互联网为主的大数据真实文本的出现,自然语言处理和文本挖掘技术也有了突飞猛进的发展。拿汉语来说,其信息处理技术已有几十年的历史,也取得了许多成就,其中一个重要的方面就是浅层分析技术的研究与应用。目前,汉语分词和命名实体识别的精度不断提高,已进入实用阶段,而与此同时,信息抽取、文本分类、聚类等文本挖掘技术也取得了巨大的进展,成为大数据分析与网络信息挖掘的有效手段。在这种情况下,人们已经不满足

于对语言本身进行分析，而是希望利用语言分析的手段从大数据中挖掘一些感兴趣的抽象对象（如事件、人物、地点、机构、音乐、软件等）。大数据搜索与挖掘就是在这个应用背景下诞生的。

大数据搜索和挖掘是建立在网络搜索、数据挖掘等基础上的。虽然传统的信息检索有很长的历史（可追溯到古代），但它真正受到人们的关注还是近半个世纪的事。如何在浩如烟海的 Web 信息中快速检索和挖掘到所需信息，已成为大数据时代必须要认真面对和解决的问题；而对网络信息的搜索和挖掘，也即本书的关注点，就成为大数据分析的重点内容之一。

1.1 大 数 据

大数据这个术语的出现大概可追溯到 Apache 的开源项目 Nutch。当时，大数据——Big Data——是用来描述为更新网络搜索索引需同时进行批量处理或分析的大量数据集。随着 MapReduce 和 Google File System（GFS）的发布，从 2009 年开始，大数据才开始成为互联网行业的流行词汇，也引起了越来越多的关注。物联网、云计算、移动互联网、手机与平板电脑、PC 以及遍布各个角落的各种各样的传感器，无一不是大数据的来源或承载方。可以说，大数据就在我们的身边——从阿里巴巴、1 号店、京东商城等电子商务数据，到 QQ、MSN 等即时聊天内容，再到 Google、Bing、百度，又到社会网络与微博数据等，都在生产、承载着大数据。随着信息处理量的增大（据统计，Facebook 对某一主题进行搜索时，需在超过 50TB 的数据中迅速找出相关内容；在 eBay 存储了超过 2PB 的数据以备查询），很多传统方法（如 RDBMS、SQL 查询等）都不能直接应用在对海量数据的处理上，否则时间开销将会变得非常大。对大数据的挖掘和分析正变得愈发重要。而对 Google、百度、Bing、Facebook、网页文本、微博消息等的挖掘与分析，使人们的行为和情绪的细化测量成为可能。挖掘用户的行为习惯和喜好，从凌乱纷繁的大数据背后找到符合用户兴趣和习惯的产品和服务并对产品和服务进行有针对性的调整和优化，本身就蕴含着巨大的商机。可以说，手握大数据的公司是“站在金矿上的富翁”。虽然一些小机构或商业组织不需要对几百 TB 的数据进行挖掘处理，但有些任务的运行已超出了单机方案所提供的能力，因此也需要一些规模较小的大数据处理系统。因此，针对大数据的处理，特别是大数据搜索与挖掘，就显得非常必要。

对大数据的智能处理，常常涉及对非结构化和半结构化数据处理、自然语言处理、信息检索、数据挖掘、大规模并行处理与云平台的构建等（因为对实时大数据进行分析需要用像 MapReduce 那样的框架来向数十、数百或甚至数千的电脑分配工作）。限于篇幅，本书不对 MapReduce 及相关的开源系统 Hadoop、Nutch 等进行详细介绍，而是将重点放在对大数据的智能化信息处理与数据挖掘上，并介绍在中

文自然语言处理、网络信息预处理、网络情报挖掘(包括网络语言分析、新特征语言抽取、汉语词法分析、文本自动分类、自动聚类、自动摘要、关键词抽取)、网络精准搜索(包括信息检索模型、句子搜索)、任务搜索等方面的研究成果,并介绍其在人物搜索与追踪中的应用。

从大数据这座金矿中挖掘有价值的信息,是本书的目的所在;而如何在云平台下组织大数据、搭建并行编程框架,则超出了本书涵盖的范围(本书只在1.2节简单介绍云计算及Hadoop相关技术),有兴趣的读者,可以参阅其他相关文献。

1.2 云计算及Hadoop简介

随着大数据时代信息量的快速增长,其对计算能力的需求远远超出自身IT架构的计算能力,这就需要不断加大系统的硬件投入,从而实现系统的可扩展性。由于传统并行模型应用的局限性,客观上要求一种易使用、易部署的并行编程框架。

从历史发展来看,网格计算的实现思路主要是通过将工作任务分发到一系列的计算机集群中,通过一些共享的文件系统等协同工作,对于要求具有繁重计算任务的程序来说,它往往具有良好的效果,但当计算需要随时读写大量数据时,其瓶颈就显现出来了。另外,它部署任务时的灵活性不够。为了节省成本和实现系统的可扩展性,云计算的概念被提了出来。

云计算是分布式计算、并行处理和网格计算的进一步发展,是基于互联网的计算,能够向各种互联网应用提供硬件服务、基础架构服务、平台服务、软件服务、存储服务。它是整个IT产业发展的趋势,也是新一代互联网、物联网和移动互联的引擎和基础。云计算与网格计算有许多相似之处,也是利用大量的计算机,构建出强大的计算能力。但云计算有更为宏大的目标,它希望能利用并发计算能力,构建稳定而快速的存储及其他服务。通常,云系统由第三方拥有的机制提供服务,用户只关心云提供的服务,而无需关心服务所在的具体位置。目前,美国政府已经建立了多个政府云。除此之外,欧盟、英国、日本、加拿大也都启动了相应的计划。我国也将云计算纳入"十二五"规划之中,并推出系列措施推动云计算的研发和应用。目前,包括Google、微软、IBM、Amazon、EMC、惠普等在内的许多IT业巨头都宣布要在云计算上进行重点研究,也有了一些供企业使用的云计算案例——如Google和IBM联合宣布推广云计算的计划等,包括卡内基梅隆大学、斯坦福大学、加州大学伯克利分校、华盛顿大学、麻省理工学院等在内的许多高校也都参加了这项计划。Amazon也提供了著名的EC2服务,这项服务可看成是一个云计算系统,能够为用户提供大规模的计算能力并有效降低响应时间。

得益于FaceBook、阿里巴巴、腾讯等行业领先企业的现身说法,Hadoop被看

成是大数据分析的利器。提到大数据的价值，就不得不谈谈 Hadoop。Hadoop 是一个由 Apache 基金会开发的开源的分布式系统基础架构，实现了 Google 的 MapReduce，主要包括 Map/Reduce 和 HDFS 文件系统。其中，主从式 HDFS 文件系统位于下层，主从式 Map/Reduce 处理位于上层，基本思想是“计算向存储迁移”——计算时各节点读取存储在自己节点的数据并进行处理。Map 处理过程一般没有数据传输，只是在 Reduce 过程中向主节点传送计算结果，有效地解决了并行计算中网络数据传输的瓶颈问题。用户可以在不了解分布式底层细节的情况下，开发分布式程序，充分利用集群的威力，完成高速运算和存储。一般地，HDFS 集群由一个名字节点（是一个管理文件的命名空间和调节客户端访问文件的主服务器）以及数据节点组成，其内部处理机制是将一个文件分割成一个块或多个块，这些块存储在一组数据节点中，名字节点操作文件命名空间的文件或目录操作，同时确定块与数据节点的映射；数据节点负责来自文件系统客户的读写请求，同时执行块的创建、删除和来自名字节点的块复制指示等。一般地，Hadoop 主要由以下几个子项目组成（刘鹏，2011）。

（1）Hadoop Common：支撑 Hadoop 的公共部分，包括文件系统、远程过程调用（RPC）和序列化函数库等。

（2）HDFS：提供高吞吐量的可靠分布式文件系统，是 GFS 的开源实现。

（3）MapReduce：大型分布式数据处理模型，是 Google MapReduce 的开源实现。

与 Hadoop 直接相关的配套开源项目一般包括以下几个方面。

（1）HBase：支持结构化数据存储的分布式数据库，是 Bigtable 的开源实现。

（2）Hive：提供数据摘要和查询功能的数据仓库。

（3）Pig：是在 MapReduce 上构建的一种高级的数据流语言，可以简化 MapReduce 任务的开发。

（4）Cassandra：由 Facebook 支持的开源高可扩展分布式数据库，是 Amazon 底层架构 Dynamo 的全分布和 Google Bigtable 的列式数据存储模型的有机结合。

（5）Chukwa：一个用来管理大型分布式系统的数据采集系统。

（6）ZooKeeper：用于解决分布式系统的一致性问题，是 Chubby 的开源实现。

1.3 Web 搜索、全文索引与 Lucene 简介

Web 搜索工具已经成为人们通过网络获取信息的必备工具。作为大数据应用中的重要成员，Web 搜索也已成为互联网中的基础应用，也是网民在互联网中获取所需信息的重要工具，其用户规模仍会随着网民总体规模的增长而进一步提

升。表1.1是中国互联网络信息中心在2013年1月发布的《第31次中国互联网络发展状况调查统计报告》中统计的2011～2012年各类网络应用使用率情况，从中可以看出，截至2012年年底，我国搜索引擎用户规模为4.51亿，较2011年年底增长了4370万，年增长率10.7%，在网民中的使用率为80.0%。

表1.1 2011～2012年各类网络应用使用率情况统计

	2012年		2011年		
应用	用户规模/万	网民使用率/%	用户规模/万	网民使用率/%	年增长率/%
即时通信	46 775	82.9	41 510	80.9	12.7
搜索引擎	45 110	80.0	40 740	79.4	10.7
网络音乐	43 586	77.3	38 585	75.2	13.0
博客/个人空间	37 299	66.1	31 864	62.1	17.1
网络视频	37 183	65.9	32 531	63.4	14.3
网络游戏	33 569	59.5	32 428	63.2	3.5
微博	30 861	54.7	24 988	48.7	23.5
社交网站	27 505	48.8	24 424	47.6	12.6
电子邮件	25 080	44.5	24 578	47.9	2.0
网络购物	24 202	42.9	19 395	37.8	24.8
网络文学	23 344	41.4	20 268	39.5	15.2
网上银行	22 148	39.3	16 624	32.4	33.2
网上支付	22 065	39.1	16 676	32.5	32.3
论坛/BBS	14 925	26.5	14 469	28.2	3.2
旅行预订	11 167	19.8	4207	8.2	—
团购	8327	14.8	6465	12.6	28.8
网络炒股	3423	6.1	4002	7.8	−14.5

由于大数据具有非结构化、海量、异构、冗余等复杂性，以及大数据挖掘与中文自然语言处理学科本身特有的复杂性等因素的共同影响，人们对大数据搜索结果的满意度还不是很高。因此，人们对大数据搜索与挖掘技术提出了新的、更高的要求，希望相应的处理系统能更加智能化、人性化，检索速度更快。这就对大数据搜索和挖掘技术提出了新的、更高的要求。

1.3.1 Web搜索

Web搜索，亦称网络信息检索，其理论基础是信息检索技术。信息检索是对信息按照一定的方式组织、存储，从海量的大数据中找到满足用户信息需求的知识，尤其指对非结构化或半结构化文本的检索（在这一点上，信息检索与基于数据

库的检索不同。在数据库领域，数据之间有特定关系并按照这种关系进行结构化存储，检索时可按照这种逻辑关系直接找到需要的信息），其主要目的是研究如何从海量文档集中高效检索出与用户需求相关的文档，其研究涉及海量信息采集、表示、组织、内容分析与知识挖掘、索引、访问、表现等方面。

传统的信息检索技术随着互联网的日渐普及，正发挥着越来越大的作用，因为在互联网中，目前非结构化和半结构化信息的队伍日趋庞大（据统计，半结构化和非结构化数据占整个信息量的80%以上）。半结构化数据的特点是有一定的结构但语义不十分确定（如Web网页中的Tag标签能够指示一些不全面的结构化信息）。非结构化信息（如大规模文本信息、语料库等）多指用户感兴趣的信息未经整理，而是分散在文本的各处，用户必须通过阅读并理解文本的方式，才能获取和理解这些信息。非结构化信息一般无法用统一的结构来明确表示，因此较难按照统一的模式进行信息结构化和信息抽取，这一点是和基于数据库的信息组织与管理有根本不同的，其检索方法主要关心文本内容及语义理解与分析，这也是网络搜索技术关注的重点和实现中的难点。

目前流行的搜索引擎是一种网络信息检索（或称网络搜索）服务。在网络信息检索中，常把Web页面看成是某种非结构化或半结构化的文档。尽管Web页面中也存在一些结构化信息，但大部分结构信息是为“显示”而设置的，它们与内容和语义无关，因此通常不把Web页面看成是有结构的。用户需要将信息需求转换为搜索引擎系统能够处理和理解的查询（可能是一组描述了用户信息需求的关键词）。搜索引擎的目标就是检索或挖掘出可能对用户有用或者相关的信息（这一点和数据库检索有很大不同），并按相关度由高到低的顺序（这一点也和数据库检索有很大不同），将检索结果返回给用户。

看似简单地搜索引擎的背后涉及包括数据结构、索引、算法、知识表示、自然语言处理、信息检索、人工智能、计算机网络、分布式处理、数据库、数据挖掘等和大数据处理相关的多个方面内容。一般地，搜索引擎主要包括信息采集、信息加工、信息检索与检索结果提供这几个部分。其中，信息采集模块以一定的策略在因特网等信息源中采集相关信息；信息加工主要指对网页资源进行信息抽取与去噪、内容分析（包括文本分析、分词处理、主题词抽取）、建立倒排索引、根据内容分析的结果编制摘要、完成信息分类等；信息检索模块则根据用户的检索提问对检索项与索引项进行匹配运算以获取对应的检索结果集，有些系统为方便用户使用还提供高级检索功能、支持自然语言提问等；检索结果提供则是在进行必要的相关分析后以超链等形式给出检索结果。图1.2为搜索引擎总体架构图。

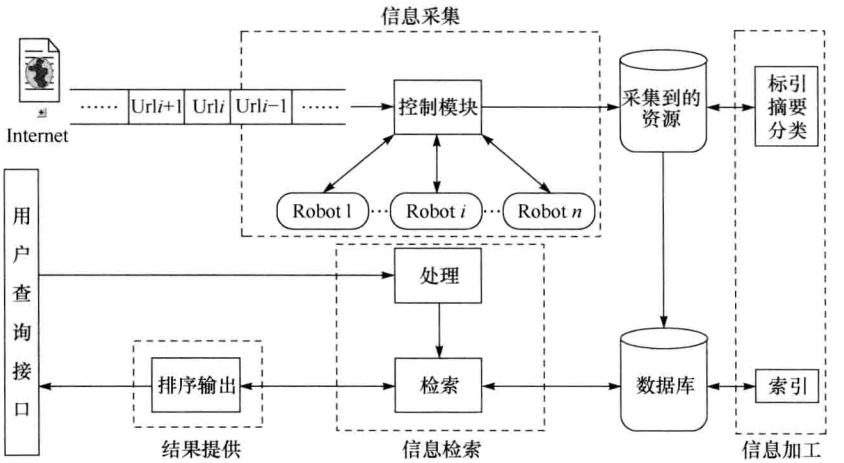


图 1.2　搜索引擎总体架构图

1.3.2　全文索引

由于关系型数据库自身结构的缘故，用其来管理大量半结构化和非结构化数据显得有些不方便(高凯，2011)。对于管理半结构化和非结构化的数据而言，采用全文检索技术是比较适宜的。全文检索包括全文索引和信息检索，前者是后者的基础。和基于结构化数据的数据库管理技术和手段不同，全文索引是指计算机程序通过扫描半结构化和非结构化文档中的每一个词，对它们建立索引并指明该词在文章中出现的次数和出现位置。当用户执行检索时，全文检索程序根据事先建立的全文索引进行检索，并将检索结果反馈给用户。经过多年的发展，全文检索从最初的字符串匹配程序已逐步演进到能对超大文本、语音、图像、多媒体等非结构化数据进行综合管理。与传统的关系型数据库相比较，全文检索有以下特点。

(1) 方便灵活的全文存储和管理功能。

(2) 支持丰富全面的检索方式，包括布尔检索(如与、或、非等)、位置逻辑检索(同段、同句、相差几个字以及前后次序有关)等几十种检索方式，可对文中任何字、词、句进行检索，还可表示检索词间的复杂位置关系。

(3) 包含的信息是原始的，库中信息基本上是未经信息加工的原始文本语料。

(4) 所用检索语言的自然性，可以以自然语言进行检索。

(5) 支持多种字符编码方式，如 GB 国标码、GBK 大字符集码、BIG5 繁体字码等。

(6) 支持多种格式文档资料的管理和检索，包括各种格式化的文档以及 HTML页面的超文本链接，自动索引格式化的文档和页面。

(7) 可以采用数据和索引压缩技术来提高系统的查询效率。

全文检索常用于对非结构化数据,即没有标记的文本进行检索,而数据库多用于对结构化数据,即有特定字段(如姓名、学号、性别等)的信息,进行检索。两者在处理的数据对象、模型、主要数据结构、查询语言上都有诸多不同之处,它们的主要区别如表 1.2 所示。

表 1.2 基于数据库的检索方式和全文检索方式的主要区别

比较项目	对结构化数据的数据库检索	对非结构化数据的全文检索
处理的对象	数据表中的记录	非结构化的文档
模型	关系数据模型	向量空间模型或其他
主要数据结构	数据表	倒排索引表
查询语句	SQL	自然语言检索

总的来说,全文检索有对检索内容进行匹配的算法,且检索结果可按匹配程度的大小从高到低排序(如向量空间模型 VSM 就是计算两篇文档之间相似度的,并据此对检索结果进行排序);数据库检索时一般没有匹配度算法,因此无法按匹配程度对检索结果排序。另外,全文检索是利用倒排索引机制来检索全文的,当数据量很大时,基于倒排索引机制的全文检索通常会非常快。

在对相关信息(包括网络蜘蛛下载的网页信息)进行信息内容分析和相关信息抽取后,需要为它们建立高效的索引,以便可以有效地访问这些信息。倒排索引不需要扫描整个文档集,而是对文档进行预处理,将文档切分成若干词(term)。索引中的每一个词,对应一个倒排表记录用来描述该词在哪些文档、文档中的哪些位置出现。图 1.3 所示的倒排索引中有 n 个词。

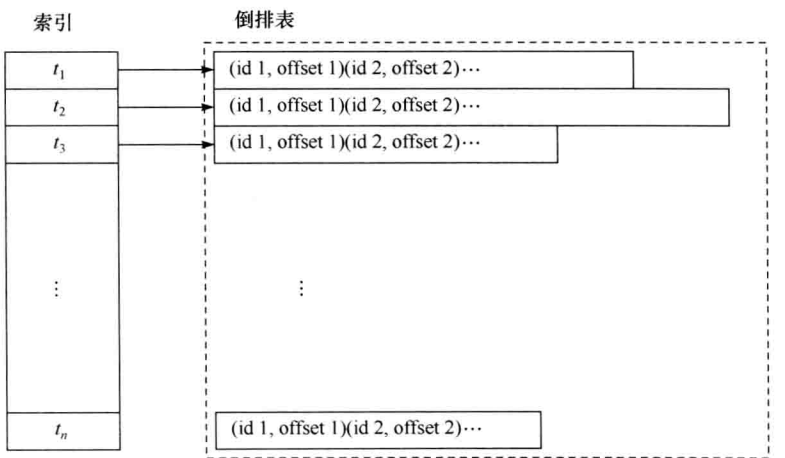


图 1.3 倒排索引示意图

倒排表中的每一个条目也可以包含词在文档中的位置信息，如词位置、句子、段落，这样的结构有利于实现邻近搜索。另外，每一个条目还可以包含词在该文档中的权重值，这个值既可以人工分配，也可以自动生成，该权重经常用于对文档的相关性计算。当所有文档的相关度计算完成后，可以根据相关度对文档排序，然后按照某种相似度算法，识别出所有与给定查询相关的文档，最后呈现给用户。

需要注意的是，构建索引时需要额外开销，因为它往往需要扫描整个文档集，将高效的倒排索引存储到辅助存储器上也需要大量的 I/O 操作。倒排索引能大大减少检索时所需的数据输入输出量。当接收到一个查询时，先查询索引，检索到相应的倒排表，最后检索算法根据倒排表的内容对文档进行排序。用户查询接口在获取用户的查询请求后，会将它转发给信息检索模块，后者则根据检索项在索引库中检索匹配的内容，其中数据处理子模块允许用户指定检索条件、相近关系等，而检索子模块则根据具体的算法在索引库中执行检索。结果提供模块在对检索结果进行相关度计算及排序后，通过查询接口将检索结果返回给用户。有的查询接口还可将用户检索行为信息（如用户使用的查询项、点击页面的 URL、停留时间等）记录到日志文件中（高凯，2010）。

1.3.3 Lucene 简介

在开发 Web 搜索应用程序时，可以借助一些开源的能够提供对海量信息索引和检索服务的平台，如 Lucene、Sphnix 等。限于篇幅，这里仅对 Lucene 进行简介。有关 Lucene 的使用、相关研究等参见作者其他相关著作（高凯，2010）。

作为目前流行的基于 Java 的开源全文检索工具包，Lucene 并不是一个完整的搜索引擎应用程序，只是为用户的应用程序提供了索引和检索功能。它提供的用于索引和检索的 API 可以方便地嵌入到各种应用中，并为具体应用领域提供全文检索。

因特网上包括各种类型的文件（如 HTML、PDF、DOC、XLS、XML 等），获取原始文本信息并不是 Lucene 的任务。在得到这些文件后，对它们进行内容分析、建立索引、完成检索等才是 Lucene 的工作。相对于数据库应用系统来说，使用 Lucene 完成全文检索具有一定的优势。首先，Lucene 可以处理经过预处理后的多种数据源，如 HTML、PDF、DOC、数据库中的数据表等；其次，Lucene 面向全文检索的优化表现在首次检索后，并不把所有的检索结果读取出来，而只将所有结果中匹配度较高的部分结果放到结果集中缓存，即使检索匹配总数很多，Lucene 的结果集占用的内存空间也不会很多。

Lucene 随着其发布版本的不同，其组成包也不尽相同，它们在系统中各自的作用如图 1.4 所示。在核心组件中，org.apache.Lucene.store 包着眼于底层的 I/O/存储结构并提供对索引存储的支持，其中的 FsDirectory 和 RAMDirectory 类

是 Lucene 中重要的两种索引方式；org. apache. Lucene. index 包完成索引。在对外接口组件中，org. apache. Lucene. search 包完成对索引数据的检索；org. apache. Lucene. analysis 包负责完成对索引和检索时检索项的分词、语言分析、过滤等功能；org. apache. Lucene. queryParser 包对用户的输入内容进行分析（如可确定检索关键词间的逻辑关系等）。在基础结构封装组件中，org. apache. Lucene. util 包中为一些公用工具类；org. apache. Lucene. document 对 Document 和 Field 的各种操作提供支持。

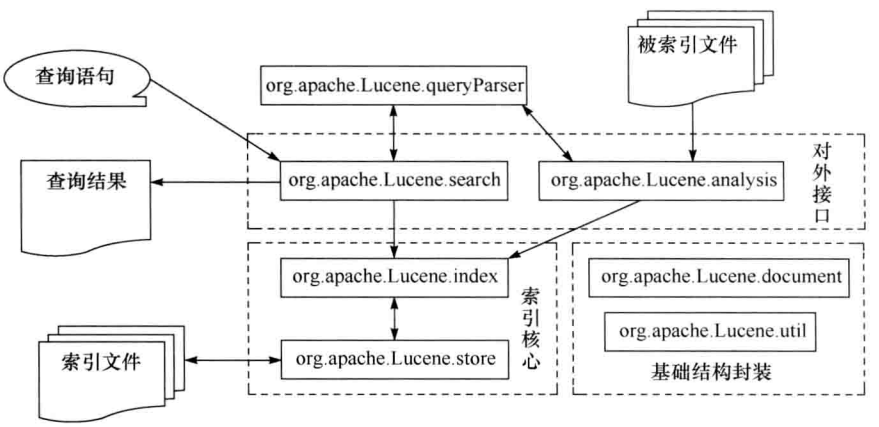


图 1.4 Lucene 主要包及其组织

分词、索引、检索处理在信息检索中占有重要的地位。分词的准确与否直接影响到后续检索。Lucene 提供了分析器用于处理词汇，此外也有一些开源的分词处理模块可供方便地集成到信息检索系统中。Lucene 的索引是利用预设的索引项字典建立按索引项排列的链表来对文件进行索引，它提供了多种不同的建立索引方式，并提供了添加、删除、更新、管理索引文件的机制。一般来说，完成信息检索任务要经过如下几个步骤（高凯，2010）。

（1）文档解析：这个任务一般由相应的解析工具而非 Lucene 来完成。

（2）内容分析：由于 Lucene 采用倒排索引作为其索引数据结构，因此 Lucene 必须先将文本信息切分成一个个的信息单元（即 Token），再利用这些 Token 构造倒排索引。在 Lucene 中，文本内容分析（如分词）的工作是由 Analyzer 完成的。

（3）生成索引：通过使用 IndexWriter 类可将文档转换成 Token，并将 Token 写入到倒排文件中。在内容分析时，已经获得了一个 Token 在某个文件中出现的频率信息、位置信息等，所以只需将这些信息按照倒排文档的格式写入到文件中即可。在生成索引时，Lucene 还有可能使用一些压缩操作等。

（4）处理用户提交的检索词：通过使用 Query 和 QueryParser 类来处理用户

提交的检索词，可完成模糊查询、语义查询、短语查询、范围查询、组合查询等多种检索操作。

(5) 完成检索：检索主要由 IndexSearcher 完成。Lucene 提供多种构建 IndexSearcher 对象的方法，如基于 Directory 的对象构建及基于文件系统的路径构建。

(6) 得到检索结果：检索结果集由 Hits 返回，通过 Hits 能获得与输入检索项相关的内容。

1.4 大数据挖掘

大数据挖掘是伴随着互联网的普及应用和传统信息检索技术的不足提出并发展起来的。大数据挖掘是从大数据中发现有用的模式，其数据可以存放在数据库、数据仓库或其他信息库中。它旨在解决数据挖掘、信息检索、知识抽取以及更广泛的商业问题。面向大数据的挖掘比面向数据库和数据仓库的数据挖掘要复杂，因为大数据往往是无结构的，通常是用长的句子或短语来表达文档类信息；有些则可能是半结构化的，当然也包括大量的异构信息和冗余信息等，对如广告、导航条、动画等无关信息的甄别与处理也都是需要考虑的问题。大数据挖掘也是一个交叉学科，它涉及信息检索(信息检索可以看成是大数据挖掘的初级阶段)、人工智能、机器学习、概率论以及数据库等。在大数据搜索和检索中，常常需要对结果进行处理和内容挖掘。应用数据分析与挖掘方法，可以帮助人们从海量网络信息中提取知识，为访问者、站点经营者以及包括电子商务在内的基于因特网的商务活动提供决策支持(毛国君，2007)。由于大数据的海量、冗余、异构等复杂特点，给传统的数据挖掘技术提出了很多亟待解决的难题。

作为大数据挖掘领域中的一个重要研究分支，文本挖掘与处理主要研究从非结构化及半结构化的海量 Web 文本信息中挖掘有价值的信息，处理其中可能隐藏的概念及其联系，并完成可能的知识发现。

文本挖掘与处理的主要对象是自然语言。自然语言的多样性和复杂性决定了计算机往往很难实现对自然语言的完全理解。在语言的深层挖掘上，目前的语言学研究和计算语言学研究尚不能充分地理解语言的内在含义，而精细语言分析也往往不能充分解决复杂的现实问题。因此，浅层语言分析可能是目前可行的一条研究路线。一方面，它能够针对需求抽取语言表层之下特定的关键信息；另外一方面，它兼顾了分析结果的准确性和系统的实用性。本书涉及的文本挖掘与处理技术也多是以某种浅层语言分析技术分析以自然语言表示的文本信息，从中挖掘出诸如新特征词、有意义串、关键词等，并在此基础上完成对文本信息的信息抽取、分类、聚类、检索、话题跟踪等处理工作。在对以自然语言表示的文本进行挖掘与处

理时，要使用自然语言理解与处理、信息检索的相关技术。计算机需要对自然语言表达的内在内容进行建模，转化为可计算的数据结构，尽可能地进行表层的理解并进一步提取其深层的语义信息。但自然语言的极端复杂性也给利用计算机来完成这种处理工作带来极大的困难。

1.5 本书主要内容及其知识点

大数据搜索与挖掘，特别是 Web 搜索与挖掘通常采用数量众多的网页搜集器和分布式的采集策略，大规模地从互联网上采集信息，然后在后台对采集的信息进行信息抽取、去噪、分类、摘要等处理工作，从而将信息加工成便于人们检索和使用的形式，并在本地服务器或集群上为这些数据建立索引。当用户检索时，系统能够按照用户的检索要求，从索引库中找到用户需要的信息并反馈给用户。但是目前的 Web 搜索系统只能为大众提供通用的信息检索服务，而且提供的 Web 搜索与挖掘服务在准确性上还远不能满足人们的需求。为此，本书将对 Web 搜索与挖掘中涉及的相关内容进行阐释，并将作者在科研工作中的心得和体会与读者分享，以期能为提高大数据搜索和挖掘系统的性能贡献绵薄之力。虽然作为交叉学科的大数据挖掘包括 Web 页面内容挖掘、页面间结构挖掘、用户访问信息挖掘、电子商务信息挖掘等，涉及人工智能、信息检索、机器学习、数据库技术等，但限于篇幅，本书只对文本挖掘相关内容进行探讨。

大数据挖掘中和本书相关的内容综述，如文本分析与挖掘、自然语言处理、分词、未登录词处理、词典组织、有意义串挖掘与主题词提取、文本内容分类与聚类等将在第 2 章进行综述。

第 3 章介绍大数据搜索中的分词处理及相关技术，并对分词对中文信息检索的影响、适用于大规模中文信息检索的分词算法的特点和算法等进行叙述。

第 4 章为浅层词法分析及其应用，重点对基于层次隐马尔可夫模型 HHMM 的汉语浅层分析及其应用、基于层次隐马尔可夫模型的汉语浅层语言分析框架、基于单字位置成词概率识别未登录词的算法等进行较详细的介绍。

面对互联网海量信息中每日涌现的新词汇，任何词典都不会囊括所有的网络新词，这不仅会给分词、词性标注等带来不利影响，也会给进一步的信息处理（如分类、聚类）等带来麻烦。为此，第 5 章给出从大数据中发现新特征语和有意义串的方法，它不仅可用于用户专业词典的编撰，还可进一步提高分词系统的准确度；在基于上下文邻接分析和语言模型分析相结合的有意义串提取算法部分，采用 N 元递增方法发现重复串，进而得到有意义串的候选串；通过计算字符串的邻接种类、邻接熵、邻接对、邻接对熵等邻接特征量，来衡量候选串是否满足语用多样性；上下文邻接分析阶段计算字符串的邻接类别等值，用来表示字符串上下文环境的丰富

程度；用熟语料上训练好的双字耦合度（来描述字与字之间结合的紧密程度）和单字位置成词概率来判断候选串的语义完整性；语言模型分析阶段计算字符串的成词概率（表示字符串的凝固性）；在基于局部性分析与伪相关反馈模型的有意义串挖掘算法中，指出局部性分析模块能有效地提取低频有意义串；伪相关反馈模型能消除分词不当（多表现为语料的应用领域性不当），而对有意义串提取带来的负面影响，具体表现为它能部分消除分词不当对有意义串提取的影响；最后，将二者相结合可以从各种语料中有效地提取有意义串。

第6章给出基于有意义串挖掘的大数据文本聚类与分类方法，通过统计分析有意义串在长度、词性以及组合方式等方面的分布规律，从大规模数据中自动分析出热点事件，对基于关键词提取的搜索结果聚类、基于K-means算法的有意义串主题聚类算法、基于邻接串种类的有意义串语境聚类等进行说明。

第7章是基于关键词提取和面向主题的自动摘要，能够对单篇或多篇文章自动提炼出内容的精华，方便用户快速浏览文本内容，可用于精化阅读、语义查询和快速匹配等。在基于关键词提取的自动摘要技术中，通过文本预处理、去除停用词、分词和用户主题词表、关键词提取、句子切分、降低句子冗余等方法完成摘要生成。在面向主题的摘要部分，通过对最大边缘相似度方法进行改进，使用倒排链表结构统计词或句子的特征，通过领域主题词表或者用户主题词的加载使用，提出句子之间包含度的概念，对文章中具有包含关系的一类句子进行有效的去冗余处理，以提高文摘质量。

作为应用，在第8章给出作者团队自行研发的JZSearch的技术方法和原理，它能完成对结构化数据库与非结构化文本相结合的搜索，支持文本、数字、日期、字符串等各种数据类型、多字段的高效搜索，支持AND/OR/NOT以及NEAR邻近等查询语法，支持维语、藏语、蒙语、阿拉伯、韩语等多种少数民族语言的检索，可无缝地与现有文本处理系统与数据库系统融合。

第9章是有关句子检索与新颖性监测，讨论句子检索的主要建模方法、句子级别的新信息检测、算法，并进一步探讨在监督学习条件下如何利用机器学习方法自动调整参数等。

第10章和第11章是相关技术在人物搜索中的应用，讨论人物搜索中的数据预处理，人物模型的组织与基于事件的信息组织，同时介绍人物追踪细粒度挖掘人物搜索引擎的设计方案，对人物对象挖掘的同一性判断问题和数据融合问题进行论述，人物模型的组织（同一性判别和动态填充）为人物追踪打下基础；基于事件的信息组织把人物参与的事件信息有效组织，能够建立出人物的重大事件信息表和活动报道专集。

1.6 本章小结

本章对大数据搜索，特别是信息检索技术、搜索引擎架构、全文检索、倒排索引等进行了简介，并对 Web 挖掘进行了简介。上述“引子”可以作为“金钥匙”，为我们开启大数据搜索与挖掘（特别是文本挖掘与分析）知识宝库的大门。

相关工作与扩展阅读

作为大数据的处理平台，刘鹏（2011）、陆嘉恒（2012）在文献中介绍了 Hadoop 的核心技术和扩展技术，包括 HDFS、MapReduce、HBase、Hive、Pig、Cassandra、Chukwa、ZooKeeper 等，并给出三个完整的 Hadoop 云计算综合应用示例。作为大数据检索的基础，孙建军和成颖（2004）系统介绍了信息检索的原理与技术，内容包括信息检索的布尔模型、向量空间模型、概论模型、逻辑模型，以及文本处理中的自动分类、自动聚类、自动文摘，设计信息检索中的查询扩展与精化、相关性、Z39.50、搜索引擎等。李晓明等（2005）系统介绍了网络搜索引擎的工作原理、实现技术及其系统构建方案，详细地讨论了大规模分布式搜索引擎系统的设计要点及其关键技术，介绍了面向主题和个性化的 Web 信息服务。Aggarwal 等（2011）则给出利用社会网络计算方法对海量数据进行分析的研究综述。

传统的数据挖掘与文本分析技术是大数据挖掘的基础。毛国君等（2007）系统阐述了数据挖掘和知识发现技术的产生、发展、应用以及相关技术、原理和算法，对数据挖掘中的主要技术分支，包括关联规则、分类、聚类、序列、空间以及 Web 挖掘等进行了理论剖析和算法描述。Han 等（2006）则在对数据挖掘相关理论、方法进行详尽介绍的基础上，展示了该领域新的研究成果，如挖掘流、时序和序列数据以及挖掘时间空间、多媒体、文本和 Web 挖掘等。Soman 等（2006）全面介绍数据挖掘的原理、方法和算法，主要内容包括数据挖掘的基本概念、数据挖掘算法的数据类型、输入和输出、决策树、数据挖掘的预处理和后处理、关联规则挖掘、分类和回归算法、支持向量机、聚类分析及多维数据可视化等。刘挺等（2008）对信息检索及信息检索系统的基本概念、原理、算法进行了介绍，包括信息检索模型、文本操作技术、文本索引和搜索技术、查询处理与 Web 检索技术、分布式信息检索、文本分类与聚类、信息过滤等，并给出了 Web 信息检索的实现实例。Buttcher 等（2012）从多个视角对信息检索技术进行了深入讲解，内容涵盖信息检索系统的架构、基础技术、词条和词项、静态和动态倒排索引、查询处理、索引压缩技术、概率模型、语言模型、分类和过滤、融合和元学习、评价方法以及并行检索、Web 检索和 XML 检索等具体应用。另外，部分有关信息检索的国际学术会议如下，在每届会议发表的论文集中，有该领域最新研发成果。

(1) TREC:http://trec.nist.gov/。

(2) Cross Language Evaluation Forum (CLEF): http://www.clef-campaign.org/。

(3) SIGIR:http://www.sigir2013.org/。

(4) CIVR:http://www.civr2013.org/。

(5) CIKM:http://cikm2013.org/。

(6) WWW:http://www2013.org/。

(7) JCDL:http://www.jcdl2013.org/。

参考文献

高凯，郭立炜，许云峰．2010. 网络信息检索技术及搜索引擎系统开发．北京：科学出版社.

高凯，张雪梅，倪素虹，等．2011. 数据库原理与应用．北京：电子工业出版社.

李晓明，闫宏飞，王继民．2005. 搜索引擎——原理、技术与系统．北京：科学出版社.

刘鹏．2011. 实战 Hadoop——开启通向云计算的捷径．北京：电子工业出版社.

刘迁，焦慧，贾惠波．2007. 信息抽取技术的发展现状及构建方法的研究．计算机应用研究，24(7):6-9.

刘挺，秦兵，张宇，等．2008. 信息检索系统导论．北京：机械工业出版社.

陆嘉恒．2012. Hadoop 实战．北京：机械工业出版社.

毛国君，段立娟，王实，等．2007. 数据挖掘原理与算法．北京：清华大学出版社.

孙建军，成颖．2004. 信息检索技术．北京：科学出版社.

Aggarwal C C. 2011. Social Network Data Analytics. New York:Springer Press.

Buttcher S, Charles L A C, Gordon V C. 2012. Information Retrieval: Implementing and Evaluation Search Engines. Massachusetts:Massachusetts Institute of Technology Press.

CNNIC. 2012. 第31次中国互联网络发展状况统计报告．北京：中国互联网络信息中心.

Han J W, Kamber M. 2006. Data Mining: Concepts and Techniques. New York:Elsevier Press.

Soman K P, Diwakar S, Ajay V. 2006. Insight Into Data Mining: Theory and Practice. New York:Prentice Hall Press.

第 2 章　大数据搜索挖掘综述

本章将对大数据搜索与挖掘中的相关技术(特别是自然语言理解与处理中涉及的相关技术)进行综述,内容涵盖文本分析与挖掘、自然语言理解与处理、中文词法分析中的分词处理、未登录词及其识别、有意义串及其识别、词典组织与管理、文本分类、文本聚类、话题识别与跟踪、信息检索的基本原理、句子级检索与新信息检测等内容。

2.1　常用的信息检索模型

当用户以一定的方式表示出其信息需求后,系统应根据用户的需求,在表示信息的数据(特别是非结构化的文本数据)中进行检索,获取与用户需求相关的结果集并按一定次序输出,对这个过程建模就产生了各种不同的信息检索模型。

一般地,一个信息检索模型要确定文档的表示方式、用户查询的表示方式及用户查询与文档间相关度的计算方法,包括文档表示、查询表示、匹配函数和结果输出等。其中,文档表示反映文档在系统中的存储形式,查询表示反映用户想要表达的信息需求,匹配函数用于把经过处理的文档表示和查询表示进行匹配计算并得到结果集,结果输出则是将检索结果集按照其和用户需求的相关性排序输出。信息检索模型是将文档、查询及其关系进行建模的框架,一般可由三元组$F(D,Q,R(q_i,d_j))$表示,其中 D 是文档逻辑视图,Q 是用户信息需求的逻辑视图,$R(q_i,d_j)$是一个与查询 $q_i \in Q$ 和文档 $d_j \in D$ 有关的函数,以便决定结果集的输出顺序(孙建军等,2004;凌云,2003;徐宝文,2003;焦玉英等,2003;Baeza-Yates et al.,1999)。

相关工作中,由 Salton 等提出的向量空间模型成功地应用于 SMART 系统中,Salton 等(1983)在 20 世纪 80 年代末又提出了扩展布尔检索模型;Wong 等(1985)建立的广义矢量模型考虑了词与词的相依性;Cooper 和 Bookstein 利用集合论建立了信息检索的一般社会模型;Maron、Roberton 和 Spark Jones 于 20 世纪 60～80 年代先后建立了三个概率检索模型;Raedchi 在模糊检索理论方面进行了研究。下面,对一些信息检索模型及常用的检索性能评价指标进行综述(高凯等,2010)。

2.1.1　传统布尔检索与扩展布尔检索模型

布尔模型是一种简单且实用的检索模型,它建立在布尔逻辑和集合论基础之

上。在传统的布尔检索模型中，文档由一组标引词描述，标引词权重只有 0 和 1 两个取值，分别表示文档中包含和不包含该标引词，而用户查询是由一组标引词构成的布尔表达式，由布尔运算符来连接运算分量，由此组成表达用户需求的提问逻辑式。检索则根据倒排索引机制找出哪些文档是与查询准确匹配的。布尔检索因其结构简单、容易实现、快速等特点广泛地在检索系统中使用，大多数的检索系统都支持布尔逻辑运算。

一般地，若 d_{ki} 表示标引词 t_k 出现了的文档，$t_k^* = \{d_{k1}, d_{k2}, \cdots, d_{kN}\}$ 表示与 t_k 有关的全体文档组成的集合，令 $\varphi(q)$ 表示对应于查询 q 的检出文档，则有(焦玉英，2003)。

(1) $q=t, \varphi(q)=t^*$

(2) $q=\neg t, \varphi(q)=\overline{t^*}$

(3) $q=t_1 \vee t_2, \varphi(q)=t_1^* \cup t_2^*$

(4) $q=t_1 \wedge t_2, \varphi(q)=t_1^* \cap t_2^*$

例如，对于标引项 t_1、t_2、t_3、t_4，假定在文档集 D 中所得的结果集如下。

(1) t_1-List：d_1, d_2, d_3, d_4

(2) t_2-List：d_1, d_2

(3) t_3-List：d_1, d_2, d_3

(4) t_4-List：d_1

对于查询 $q=(t_1$ AND $t_2)$OR $(t_3$ AND (NOT $t_4))$来说，$(t_1$ AND $t_2)$得到的检索结果为$\{d_1, d_2\}$，$(t_3$ AND (NOT $t_4))$得到的检索结果为$\{d_2, d_3\}$，因此对于$(t_1$ AND$t_2)$OR$(t_3$ AND (NOT $t_4))$的检索结果就为$\{d_1, d_2, d_3\}$。

布尔检索模型简单易用，其缺点主要是逻辑表达式较死板，检索结果不便按照相关性排序输出。为此，研究人员提出了多种改进方案。有研究人员将检索出的文档集合和用户的检索需求进行模糊逻辑比较，并按照相关性给出检索结果。相关文献(Salton et al.，1983；Lee et al.，1993)提出一种扩展布尔模型(extended boolean model)，它在保持传统布尔检索结构性的同时，吸取了模糊检索和向量检索模型的长处，不仅允许对文档向量和查询向量中的标引项加权，还允许对布尔表达式中的连接符加权，并通过一个特殊的参数 p 来控制布尔操作符的严格程度，是对传统的布尔检索和矢量处理方法的折中。限于篇幅，相关策略不再赘述，有兴趣的读者可以查阅相关文献。

尽管布尔模型存在不足之处，但这并不妨碍其在商业上的成功应用，这是因为布尔模型一般仅关心检索项在文档中出现与否，因而通过适当的索引其检索效率一般较高。

2.1.2　向量空间模型

由美国康奈尔大学的 Salton 提出的向量空间模型的基本思想是将文档库中

的每一篇文档 $d_i(i=1,2,\cdots,N)$都参照文档库相互独立的 t 个特征属性向量 $T=(t_1,t_2,\cdots,t_t)$标引成相应的 t 维文档向量 $d_i=(x_{i1},x_{i2},\cdots,x_{it})$。其中，$x_{ij}(j=1,2,\cdots,t)$是文档 d_i中具有特征属性 t_j的测度(可将其理解为词条在文档 d_i中出现频数的某种函数)，$x_{ij}=0$ 表示文档 d_i与 t_j完全无关，$x_{ij}=1$ 表示文档 d_i与 t_j完全相关。令 D 表示由诸特征属性向量组成的文档库特征属性矩阵，如果用户的检索提问参照 T 排成提问向量 $Q=(q_1,q_2,\cdots,q_t)$，并将⊙定义为相似度运算，那么通过计算相似度 $\mu_i=d_i\odot Q$，就可得到某一篇文档 d_i满足用户提问 Q 的测度。如果同时给出某个阈值 λ(或文档数 M)，则当 $\mu_i>\lambda$(或测度最大的前 M 个 μ_i)时，对应的文档被选中，可以得到相应的检索结果(王永成，1991)。这样，文档和查询间的匹配问题就转化为向量空间矢量的匹配问题。该模型主要涉及两方面工作：一是如何构建一个向量来表示文档中的词项；二是如何构建另一个向量来表示查询中的词项。我们必须选择一种方法来度量任意文档向量和查询向量的相似度。一种方法是可以比较两个向量差值的大小，然而由于查询一般比较短小，会导致较大的向量看起来往往和大多数查询都不相关。判断两个向量相近性的传统方法是计算这两个向量之间的夹角。两个向量的夹角可以通过内积(或称点积)计算出来，然而并没有必要真正计算出角度的实际值——任何计算夹角的单调函数都可以达到实用的要求。通常使用相似度来代替夹角。计算相似度有多种方法，通常来说内积效果最佳。只有文档向量和查询向量指向的方向大体一致，它们才相似。可见，向量空间模型就是把每一个标引词看作一个向量，可对每个标引词定义权重，再通过某种计算得到向量间的相似度，根据相似度的大小对检索结果进行排序。

对于文档集中每一个不同的词项(或概念)，在向量中只记录一个分量。接下来，考虑在文档集中只有两个不同词项，即 α 和 β 的情况。所有的向量只包含两个分量：第一个分量表示词项 α 的出现情况，第二个分量表示词项 β 的出现情况。构造向量最简单的方法是：当词项出现时，就在对应向量的分量处记 1；如果词项未出现，就在对应的分量处记 0。例如，文档 D_1，其中词项 α 出现了 2 次，词项 β 出现了 0 次。这个文档使用二值表示方法时，对应的向量为〈1,0〉。二值表示方法可以用来计算相似度，但是并没有考虑一个词项在文档中出现的次数。通过扩展这种表示形式，将词项在文档中出现的频率作为向量中各个分量的值。在这个例子中，向量可以表示为〈2,0〉。

除了简单地给出查询词列表外，用户通常还会给出权重，该权重表示一个词项比另外一个词项更重要。这是通过在初始查询中用户人工指定词项权重来实现的。另外一种方法是自动指定权重，通过基于词项在整个文档集中出现的频率。基本思想是不频繁出现的词的权重应该比频繁出现的词的权重更高。相关文献(Salton，1970，1969)分别采用权重自动赋值与人工赋值方法计算相似度，然后进

行查询比较。实验结果表明，自动赋值在查询性能上并不比和人工赋值差(Salton，1970，1969)。20 世纪 70 年代，研究者研究了不同文档集的权重，其结论是如果引入不同文档集的权重，能够提升相关度排序的效果。

我们采用更加形式化的定义，并采用稍大一些的例子来展示如何使用基于数据集频率的权重，对于一个给定的词项，其权重使用逆文档频率(IDF)来计算。为了给每篇文档建立一个对应的向量，有如下定义。

(1) t 表示文档集中不同词项的个数。

(2) tf_{ij} 表示词项 t_j 在文档 D_i 中出现的次数，也就是词频。

(3) df_j 表示包含词项 t_j 的文档的篇数。

(4) idf_j 表示 $\lg\left(\frac{d}{\mathrm{df}_j}\right)$，其中 d 表示所有文档的篇数。该参数就是逆文档频率。

对于每一篇文档向量，都有 n 个分量，并且对于整个文档集中每个不同的词项，都包含一个词条。向量中的每个分量为在整个文档集中计算出来的每个词项的权重。在每篇文档中，词项权重基于词项在整个文档集中出现的频率情况以及词项在某一个特定文档中出现的频率自动赋值。词项在一篇文档中出现的频率越高，则权重越大；相反，如果词项在所有文档中出现的频率越高，则权重越小。只有当词项在文档中出现时，文档向量中词项的权重才为非零值。对于一个包含许多小文档的大文档集，文档向量可能会包含大量的零元素。例如，一篇文档集包含 10 000 个不同的词项，也就是每个文档中要用 10 000 维的向量来表示。一个给定的只有 100 个不同词项的文档向量包含 9900 个零分量。

对于文档中词项的权重因素，主要综合考虑词频和逆文档频率。也就是说，使用下面的公式计算文档 i 对应的向量中第 j 个词条的值，即

$$d_{ij}=\mathrm{tf}_{ij}\times\mathrm{idf}_j$$

当我们在一篇文档检索系统中用文档集中 t 个不同的词项来查询时，系统将为每个文档计算维度为 t 的向量 $D(d_{i1},d_{i2},\cdots,d_{it})$。向量值使用前文所述的词项权重填充。类似地，查询中的词项构建的向量为 $Q(w_{q1},w_{q2},\cdots,w_{qt})$。

查询 Q 和文档 D_i 的相似度可以简单地定义为两个向量的内积。这种策略也常常被用来计算两篇文档的相似度。

向量空间模型一般用查询向量与文档向量间的夹角余弦值来计算用户查询和文档间的相似度，即

$$\mu_i = \mathrm{sim}(d_i, Q) = \frac{\vec{d_i} \cdot \vec{Q}}{|\vec{d_i}| \cdot |\vec{Q}|} = \frac{\sum_{p=1}^{t} x_{i,p} \times q_p}{\sqrt{\sum_{p=1}^{t} x_{i,p}^2} \times \sqrt{\sum_{p=1}^{t} q_p^2}}$$

其中，d_i表示文档库中的一篇文档，由 t 个标引项进行表示；x_{ij} 为相应的权值，用户的查询由向量 Q 来表示(Salton，1971)。

向量空间模型的优点之一是可以对查询向量中的词条赋予权值，因而便于通过相似度对检索结果进行排序。但是，该模型认为和查询最相关的文档是那些在用词规律上和查询类似的文档，这种强加的序化不尽合理。此外，文档之间的相似度用向量内积或余弦测度来表示，而这种方法的理论前提是词条间应相互独立，即每个词条在一篇文档中出现与否和其他词条无关。显然，这种假设忽略了词条间固有的关联性及它们之间可能存在的语义联系，因此研究人员相继提出了多种改进措施。潜在语义索引(LSI)以奇异值分解方法来降低向量模型的维度，已被用于多种实际应用系统中(包括词典获取模型、问答系统等)(焦玉英，2003)，其理论假设是文档中的单词用法包含了潜在的结构，可以通过统计学方法来估计这种结构，并且词条、文档和用户查询都可以用这种潜在的结构进行描述(Baeza-Yates and Ribeiro-Neto，1999)。文献(王永成，1993)指出了向量空间模型的缺点并提出了修正公式。Gao 等(2004)提出一种用查询矩阵来表达用户检索需求的方法。Wong 等(1985)提出了广义向量空间模型(GVSM)，在 Smart 系统中对广义向量空间模型和传统向量空间模型做了比较，结果表明广义向量空间模型的检索效率在很大程度上优于传统的向量空间模型(焦玉英，2003)。

在实现向量空间模型或其他检索模型时，通常使用倒排索引来避免在每篇文档中使用冗长的顺序查找来查找查询词。在用户提交查询之前，首先构造倒排索引表。倒排索引表的结构如图 2.1 所示。n 个词项中的每一项都被存储在称为索引的结构中。对于每一个词项，其指向的逻辑链接表被称作倒排表。倒排表记录每篇包含该词项的文档。在图 2.1 中，倒排表中记录了文档标识符和词频等信息。实际上，往往使用比链接表更有效的结构，但从概念上讲都是为了同一目的。图中的倒排表表明词项 term_1 在文档 1 中出现了 1 次，在文档 10 中出现了 2 次。

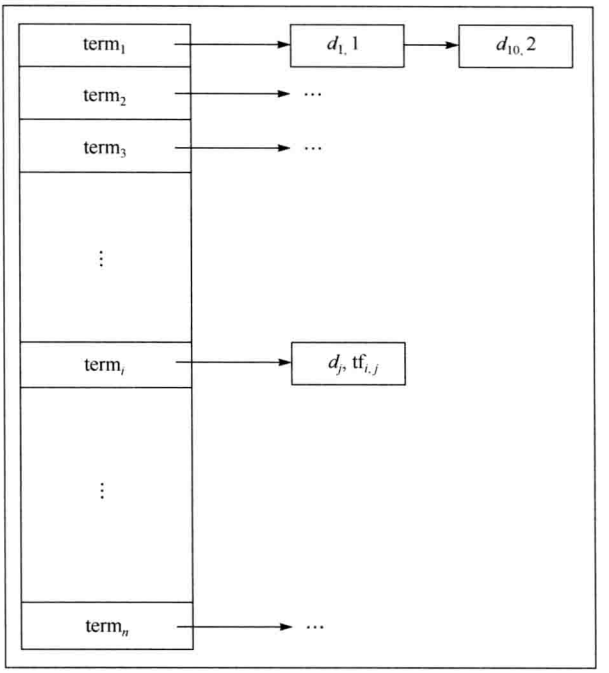


图 2.1　倒排索引

2.1.3　概率检索模型

概率论模型的基础是概率，该模型预估计信息资源与用户需求的相关性，根据相关性大小进行排序，因此排到最前面的文档将会是最有可能满足用户需求的文档。van Rijsbergen 和 Robertson 等提出的概率检索模型的基本思想是根据先前检索过程中得到的相关性先验信息来计算文档集合中每篇文档成为相关文档的概率，并根据统计理论（如贝叶斯决策等）来确定哪些文档可作为输出文档集（焦玉英，2003）。该算法基于文档集中每个词项在相关文档中出现的可能性计算出概率，对文档和查询匹配的所有词项计算联合概率，从而得出文档与查询的相似度。相关工作（Losee and Bookstein，1988）将布尔检索和概率检索模型有机地结合起来，但在没有获得样本文档之前，无法估计词条的相关性且该方法复杂度较大（徐宝文等，2003；焦玉英，2003）。

概率模型要设法解决两个基本问题，即参数估计和独立性假设。参数估计是指精确的概率计算需要基于相关性估计。如果没有良好的训练数据集，通常很难精确地估计参数。第二个问题是独立性假设。尽管在现实中，独立性假设并非完全符合真实情况，但是许多概率模型都要求这种独立性假设。限于篇幅，这里不再

对概论模型中的相关算法进行详细的介绍，有兴趣的读者，可以参考作者翻译的由Grossman所著的 *Information Retrieval: Algorithms and Heuristics (Second Edition)*（张华平等，2010）。

2.1.4 语言模型

统计语言模型是生成一段文本的概率机制。因此，模型需要定义所有可能的词序分布。最简单的语言模型是一元语言模型，只需要每个词的分布；更复杂的语言模型需要更多的上下文信息（例如词的前文）来预测下一个词（Rosenfeld，2000；Charniak，1993）。尽管使用语言模型进行语音识别和机器翻译有二十多年的历史（Hodjat et al.，2003），语言模型用于信息检索仅起源于1998年（Ponte and Croft，1998）。语言模型的核心思想是文档可以通过它们生成查询的可能性的大小来排序。考虑朗读一篇文档，如果说话人读出该文档中的所有词语，那么他再说出查询中的词语的概率是多少呢？在形式上，相似度可以简单地定义为

$$\mathrm{SC}(Q,D_i)=P(Q|M_{D_i})$$

其中，M_{D_i}是文档D_i中隐含的语言模型。

需要明确定义前面所讲的生成查询。也就是说，我们需要为查询构造一个概率模型。一种方法（Ponte and Croft，1998）是对任何词项的出现与否使用独立的伯努利事件建模，并且将生成整个查询看做观察到查询中的词项和没有观察到查询外的词项的联合事件。在这种情况下，生成查询的概率是所有查询中出现的词项和没有出现的词项的概率的乘积，即

$$\mathrm{SC}(Q,D_i)=\prod_{t_j\in Q}P(t_j\mid M_{D_i})\prod_{t_j\notin Q}(1-P(t_j\mid M_{D_j}))$$

可以通过多种方式来估计模型中的$P(t_j|M_{D_i})$，一种直接的方法如下，即

$$p(t_j|M_{D_i})=p_{ml}(t_j|M_{D_i})$$

其中，$p_{ml}(t_j|M_{D_i})$表示同项分布的最大似然估计，如相对词频，即

$$p_{ml}(t_j|M_{D_i})=\frac{\mathrm{tf}(t_j,D_i)}{\mathrm{dl}_{D_i}}$$

其中，dl_{D_i}是文档D_i的长度。

语言模型的基本思想如图2.2所示。该模型可以使用相似度，但是存在比较大的问题。如果查询中的词项在文档中没有出现，整个相似度就为零。考虑前面所用的一个查询和三篇文档的小例子。

Q："gold silver truck"

D_1："Shipment of gold damaged in a fire"

D_2："Delivery of silver arrived in a silver truck"

D_3："Shipment of gold arrived in a truck"

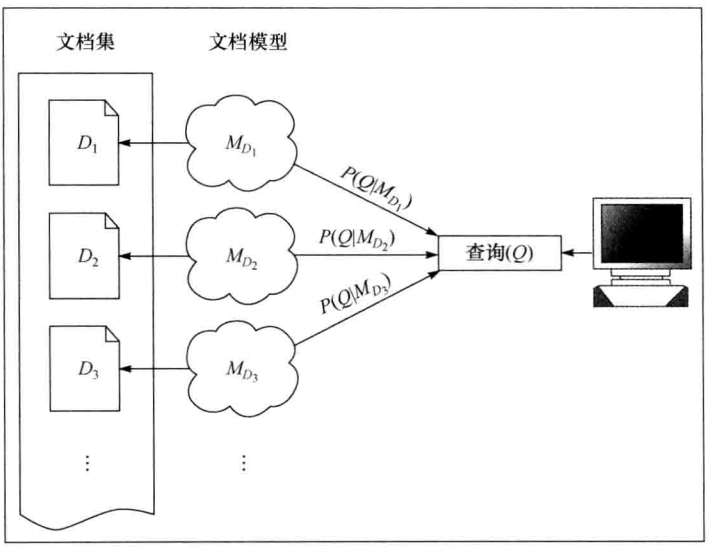


图 2.2　语言模型

词项 silver 在文档 D_1 中没有出现。类似地，silver 在文档 D_3 中没有出现，并且 gold 在文档 D_2 中也没有出现。这个例子将导致三篇文档的相似度都为零，因此最大似然估计为

$$p_{ml}(\text{silver}|M_{D_j})=\frac{\text{tf}(\text{silver},D_i)}{\text{dl}_{D_i}}=0$$

为了避免查询中的词项在文档中没有出现引起的问题，现在有许多种这些词项估计非零值的平滑方法。一种方法是假设查询词项可能在这个模型中出现，但是不会高于它在其他文档中出现的比率。最初提出比值$\frac{\text{cf}_t}{\text{cs}}$，其中 cf_t 表示词项 t 在文档集中出现的次数，cs 表示词项所有词项在文档集中出现的次数。在这个例子中，silver 的值估计为 2/22=0.091。另一种调整方法源于以下事实，即这些文档模型仅仅基于单个独立的文档。这些数据量对于构建一个模型来说非常小。研究者提出使用大的样例数据（整个文档集）来进行如下估计，即

$$p_{\text{avg}}(t)=\frac{\sum_{d(t\in d)}p_{ml}(t\mid M_d)}{\text{df}_t}$$

其中，df_t 是词 t 的文档频率，其倒数就是前文中计算的逆文档频率。

为了提高估计词项权重的效果，最小化估计风险是有可能的。定义 $\overline{f}_t$ 为文档中词 t 的平均词频，其计算公式为

$$\overline{f}_t = p_{\text{avg}}(t) \times \text{dl}_d$$

通过使用几何分布可以得到其风险率，即

$$R_{t,d} = \left(\frac{1.0}{1.0+\overline{f}_t}\right) \times \left(\frac{\overline{f}_t}{1.0+\overline{f}_t}\right)^{\text{tf}_{t,d}}$$

第一种相似度计算公式描述的是在信息检索使用的语言模型中，对查询中没有出现的词项使用平滑因子$\frac{\text{cf}_t}{\text{cs}}$；如果该词在查询中存在，则在小规模文档模型上估计 w 的值，并引入风险函数作为混合参数。词项权重的计算公式可以表示为

$$P(t|M_{di}) = \begin{cases} p_{ml}(t,d)^{(1-R(t,d))} \times p_{\text{avg}(t)}^{R(t,d)}, & \text{tf}(t,d)>0 \\ \dfrac{\text{cf}_t}{\text{cs}}, & \text{其他} \end{cases}$$

2.2 自然语言理解与处理概述

自然语言具有语言属性与自然属性。语言属性表现为公认的某些约定俗成的内在规律性，如“把车开回北京去”是一个符合这种内在规定性的构成形式，而“车回开北京把去”就是一种不符合这种内在规律性的构成形式。自然属性是说并不存在某个人为制造的、严格的语法规则体系来约定人们的语言表达方式，这和程序设计语言是大相径庭的。自然语言需要遵循一定的内在规律，但更大程度上是存在即合理。例如，“你先走”符合人们的正常使用习惯，后来随着网络的流行，“你走先”、“给个理由先”都逐渐成为了合理的语言新现象。按照相关文献中的描述，计算机对自然语言的研究和处理，大致经过如下三个过程。

（1）把需要研究的问题在语言学上加以形式化，使之能以一定的数学形式，严密而规整地表示出来。

（2）把这种严密而规整的数学形式表示为算法，使之在计算上形式化。

（3）根据算法编写计算机程序，使之在计算机上加以实现。

一个自然语言处理系统必须考虑许多语言自身与结构方面的知识，如什么是词、词如何组成句子、词的意义是什么、词的意义对句子意义有什么贡献等，但这些却还是远远不够的。例如，一个系统如果要回答提问或者直接参与对话，它不仅需要知道很多语言结构的知识，而且还要知道人类世界的一般性知识并具备人类的推理能力。因此，许多语言学家通常把对语言的分析和理解分成如下几个主要层次，即词法分析、句法分析、语义分析、篇章分析。

按照多数文献的观点，首先词法分析主要包括分词、词性标注、词义消歧、新词识别等，是通过分词、词频和位置统计等手段获得相关语言信息。其次，句法分析通过使句子成分特征化来分析句子结构特征，通过对句子和短语结构的分析找出

词、短语等的相互关系以及各自在句中的作用，并以一定的结构来表达诸如从属关系、成分关系等，目的是判定句子中各种结构性成分。第三，为了理解一个提问，一般还需要更多的语义和语用知识来帮助理解句子的意思，通过分析找出词义、结构意义及其结合意义，从而确定句子所表达的真正含义（俞士汶，2004），而语义信息的标记需要包含概念完全集与关系图的支持，需要对句法成分做出细致的语义分类，一般应包括语言层面（反映语言表面现象的知识，如同义词关系、层次关系等）、本体论层面（描述概念之间复杂的语义关系）、常识层面等。虽然这项工作浩繁，但目前已经取得了一些初步成果。最后，篇章分析用于对多个语句、段落之间在结构或者语义上的相互关系进行分析（王晓龙等，2005）。

按照多数文献中的分类方法，从技术路线上看，计算机自然语言处理可大致分为基于语言学规则分析的方法和基于统计的方法这两类主要方法（当然，也有不同的分类方法，限于篇幅在此不再赘述）。前者通过对语言学知识的形式化规则处理，将语言学知识转化为计算机可处理的形式，而后者则是从大规模真实语料库中获得统计信息，并将之用于自然语言的分析处理。基于规则分析的方法主张以建立形式化的知识系统来表述语言知识，其本质是一种确定性的演绎推理方法，如 20 世纪 60 年代 Chomsky 的转换生成语法 TG、Woods 的扩充转移网络，以及随后的 Montague 语法、格语法与语义网络理论等。多数文献又将它大致分为基于语法规则和语义规则两个方面。目前，语法和语义理论开始趋于融合，如后来发展的 TG 理论引进了语义因素，Bresnan 和 Kaplan 的词汇功能语法用一种结构来表达特征、功能、词汇和成分的顺序。虽然这些方法基本可以完成对单个句子的分析，但一般很难覆盖全面的语言现象，特别是对于段落或篇章的理解更是无从下手，同时规则之间的相容性和适用范围也存在一些限制。另一方面，自 20 世纪 80 年代以来，由于大量语料库的出现以及计算机处理能力的提高，基于规则的方法迟迟未能达到人们预期的目标，基于统计的方法逐渐兴起。该方法主张通过搜集实际语言材料形成语料库，并在此基础上进行分析和处理。它通过构造统计模型来对语料库中描述的实际出现的语言现象进行统计，进而得到统计意义上的语言知识，其本质是非确定性的基于概率的推理方式。由于知识是通过对语料库进行分析后抽取出的，因此它可为语言处理提供较为客观的依据。但基于统计的方法本质上是一种非确定性推理方法，可能会掩盖一些小概率事件。由于有些统计方法无法解决的问题利用规则方法却容易得到解决，因此以实用性为目标，将基于规则的方法与基于统计的方法有机结合起来，构建面向大规模真实文本的信息处理才是未来发展之趋势。

相关工作中，Grishman、Doszkocs、Warner 和 Smeaton 等讨论了自然语言理解在信息检索领域中的应用（焦玉英，2003）。赵军等（2005）在标引和相似度计算中引入语义计算，用浅层语义来指导检索过程。姜维等（2007）基于混合语言模型

构建了实用汉语词法分析系统。李晗静等(2007)研究了基于自然语言理解的空间关系可视化。Chen 和 Chan(2007)基于数据挖掘与粗糙集理论给出一种中文语言分析模型。Minwoo 和 Lee(2008)通过对语言环境特征和语义信息的分析,提高系统对自然语言口语识别的准确性。Bao 等(2006)提出 LSM 语言处理模型来增强系统对自然语言的处理能力。

2.3 中文词法分析中的分词处理

在中文自然语言处理中,词是最小的能够独立活动的有意义的语言成分。汉语以字为基本书写单位,词语之间没有明显的区分标记,因此进行中文自然语言处理通常是先将汉语文本中的字符串切分成合理的词语序列,然后再在此基础上进行其他分析处理。中文分词是中文信息处理的一个基础环节,已被广泛应用于中文文本处理、信息提取和文本挖掘等应用中。分词涉及许多方面的问题,主要包括如下内容。

(1) 核心词表问题:许多分词算法都需要有一个核心的(通用、与领域无关的)词表。凡在该词表中的词,分词时就应该切分出来,但对于哪些词应当收进核心词表,目前尚无一个标准。

(2) 词的变形问题:汉语中的动词和形容词有些可以产生变形结构。例如,“打牌”、“开心”、“看见”、“相信”可能变形成“打打牌”、“开开心”、“看没看见”、“相不相信”等,它们可以被切出“打打/牌”,但“开开/心”就不合理;“看/没/看见”说得过去,“相/不/相信”就说不过去。又如,大量的离合词,如“打架”、“睡觉”等可以合理地变形为“打了一场架”、“睡了一大觉”。对这些变形结构的切分往往缺少可操作而又合理的规范。

(3) 词缀的问题:如语素“者”在现代汉语中单用是没有意义的,因此“作者”、“成功者”、“开发者”内部不能切开。依据这个标准,如“作出了巨大个人财产和精神牺牲者”、“克服许多困难而最终获得成功者”、“开发中国第一个操作系统软件者”也不能切开,但这样复杂的结构与词的定义相矛盾。又如,职务名称“教育局长”,语义上理解为“教育局之长”,但切成“教育/局长”、“教育局/长”、“教育/局/长”或不予切分,都会有人提出异议。

(4) 汉语自动分词规范须支持各种不同目标的应用,但不同目标的应用对词的要求是不同甚至是矛盾的。例如,以词为单位的键盘输入系统为了提高输入速度,一些互现频率高的相互邻接的几个字也常作为输入单位,如“这是”、“每一”、“并不”、“不多”、“不在”、“就是”、“也就”等。在检索系统中,检索系统的词库注重术语和专有名词,且多数检索系统倾向于分词单位小型化,例如把“中国科学院”切成“中国/科学院”,使得无论用“中国科学院”还是用“科学院”检索,系统都能查到,

提高查全率。在校对系统中往往将含有易错字的词和词组作为词单位，如许多人“作”和“做”分不清，计算机自动判别时，若把它们当做单字词也不好区分，但在同前后文构成的词或词组中往往可以有确定的选择，故应把有关的词和词组都收进词库，如“敢做”、“敢作敢为”、“叫做”、“做出”、“看作”、“做为”等，而校对系统要求分词单位较大，如把“勇斗”、“力擒”、“智取”等分别作为一个分词单位并划归及物动词参与上下文检查，“张老师”、“五分之三”、“北京中医学院”也应分别作为分词单位，并分别归类作为人、数字、机构名，再参与上下文检查。

总的来说，汉字分词的困难可能表现在如下三个方面。

(1) 汉字切分歧义问题。汉语言本身存在歧义。对人来说，一般是通过上下文来理解，但是机器很难正确判断该如何切分。例如，对“组合成机器”来说，“组合”、“合成”都是词，到底是切分成“组合/成”还是切分成“组/合成”？歧义一般来说按照结构可以分为交叉歧义和覆盖歧义。对于字串“ABC”来说，如果“AB”、“BC”分别属于两个不同的词，它们既可切分成“A\BC”，也可切分成“AB\C”，则称“ABC”存在交叉歧义(例如在字串“组合成”中，“组合”和“合成”都分别是一个词)。对于字串“AB”来说，如果“AB”是一个词，“AB”其中的子串“A”或“B”也是一个词，则称“AB”存在覆盖歧义，如在字串“马上”中，“马上”是一个词，而“马”和“上”在某些上下文中又单独成词(如“他/从/马/上/下/来”)。歧义的消除一般需要提供更多的语法和语义信息，有时要结合上下文语境理解。例如，对于“学生会组织义演活动”来说，是切分成“学生/会/组织/义演/活动”还是“学生会/组织/义演/活动”？仅靠这一句是无法判断的，必须要结合更多的上下文才能理解。

(2) 汉语未登录词识别问题。未登录词指的是词典中没有收录的词，包括各种命名实体(如数词、人名、地名、机构名、译名、时间、货币)和网络新词(如坑爹、酱紫、快女、给力)等。另外，一些缩略语(如科协)和术语(如股骨头坏死)也属于未登录词的范围。有些命名实体(如数词、时间等)有比较强的规律性，相对容易识别；有些人名、地名、机构名等也有一些自己的常用词和规律，采用一些规则和统计信息也能识别。随着社会的发展，各种新词层出不穷，很多新词的形成在语法语义上是完全没有规律的，而且很多未登录词和常用词混在一起进而形成歧义，如“[王辉[国]家]里有点急事”、“发烧[是[非]典]的典型症状之一”等。如何识别这些不规范的新词和一些容易产生歧义的命名实体，是中文分词的一大难点。

(3) 很多算法都是将已知词识别和未登录词识别分作两个部分。这样带来的问题就是究竟先识别哪个部分更有利。如果先识别已知词，会导致未登录词识别错误(如“内塔尼亚/胡说”)；如果先识别未登录词，同样也可能导致已知词识别错误(如“胜利取决/于勇/气”)。对于哪一步识别在先更好一些，目前并没有一个确定的结论。

从 1983 年第一个实用分词系统 CDWS 的诞生到现在，国内外的研究者在汉

语分词方面进行了广泛的研究，提出了很多有效的算法。

2.3.1 基于词典和规则的汉字分词

切分时，用待切分的字符串去匹配词典中的词条，如果匹配成功，则将其切分成一个词。这类方法包括各种形态的最大匹配分词方法和全切分分词算法等。

1. 最大匹配分词方法

最大匹配分词方法又分正向最大匹配、反向最大匹配和双向最大匹配方法。正向最大匹配从左到右每次取最长词；反向最大匹配每次是从右到左取最长词；双向匹配则是进行正向、反向匹配，对于两种匹配结果不同的地方再利用一定的规则进行消歧。

最大匹配法可能无法处理部分覆盖歧义和交叉歧义，但这种方法实现简单且切分速度快。

2. 全切分分词算法

利用词典匹配，获得一个句子所有可能的切分结果。由于全切分的结果数随着句子长度的增加呈指数增长，因此这种方法的时空开销较大。对于比较长且包含较多歧义的句子，往往要经过很长时间才能遍历完所有的切分路径。

3. 基于理解的中文分词算法

分词中歧义消除的过程是一个理解的过程，不仅需要词法信息，还需要句法和语义信息。所以目前也有些研究者尝试模拟人的理解过程，在分词过程中加入句法和语义分析来处理歧义问题。由于汉语语言知识的复杂性，难以将各种语言信息组织成机器可直接读取的形式，因此目前基于理解的分词系统还处在实验阶段。

2.3.2 基于大规模语料库的统计学习的分词方法

这类方法主要是利用从大规模语料库中通过统计得到的各种概率信息对中文字符串进行切分。这种方法往往不需要人工维护规则，也不需复杂的语言学知识，且扩展性较好，是现今分词算法中较常用的做法。

1. N 元语法模型和隐马尔可夫模型

在 N 元语法中，一个句子可以看成一个连续的字符串序列，它可以是单字序列，也可以是词序列。一个句子 $W_1W_2\cdots W_k$ 的出现概率用 $P(W)$ 表示，即

$$\begin{aligned} P(W) &= P(w_1 w_2 \cdots w_k) \\ &= P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_1 w_2)\cdots P(w_n \mid w_1 w_2 \cdots w_{n-1}) \\ &= \prod_{i=1}^{n} P(w_i \mid w_1 w_2 \cdots w_{i-1}) \end{aligned}$$

不难看出，N 元语法模型认为产生语句 W 的概率是产生其第一个字 W_1 的概率，乘以在给定第一个字 W_1 的条件下产生第二个字 W_2 的概率，再乘以在给定前两个字 W_1W_2 的条件下产生第三个字 W_3 的概率，最后再乘以在给定前$(n-1)$个单字的条件下产生第 n 个字 W_n 的概率。产生第 i 个字的概率是由已产生的前 $i-1$ 个字 $w_1, w_2, \cdots, w_{i-1}$ 来决定的。其中的二元语法模型(2-gram)也叫一阶马尔可夫链，它有如下的公式，即

$$\begin{aligned} P(W) &= P(w_1 w_2 \cdots w_k) \\ &= P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_2)\cdots P(w_n \mid w_{n-1}) \\ &= \prod_{i=1}^{n} P(w_i \mid w_{i-1}) \end{aligned}$$

n 元语法的作用是预测了一个单词序列出现的概率。它假设一个单词出现的概率分布只与这个单词前面的$(n-1)$个词有关，而与更早出现的单词无关。这无疑简化了可能的计算过程。利用大规模语料库和成熟的 N 元语法模型，可以容易地得到比较高的切分正确率。已经有研究表明，使用三元语法且在不考虑未定义词的情况下，可以将切分的正确率提高到 98%以上。

隐马尔可夫模型(hidden Markov model，HMM)最先由 Baum 提出，它在自然语言理解的语音识别、词性标注等有广泛应用。在该模型中，对于一个随机事件，有一个状态序列$\{X_1, X_2, \cdots, X_n\}$以及一个观察值序列$\{Y_1, Y_2, \cdots, Y_n\}$。该模型是指输出链可以被观察到，但状态链是隐藏的。隐马尔可夫模型的作用类似于译码，就是从序列中的每个值(观察值)去推测它可能属于哪个状态。单独用隐马尔可夫模型来分词的不是很多，通常是和词性标注结合在一起用。隐马尔可夫模型可以形式化为一个五元组(S, O, A, B, π)，其中 $S=\{q_1, q_2, \cdots, q_n\}$是状态值的有限集合；$O=\{v_1, v_2, \cdots, v_m\}$是观察值的有限集合；$A=\{a_{ij}\}$，$a_{ij}=p(X_{t+1}=q_j \mid X_t=q_i)$是转移概率；$B=\{b_{ik}\}$，$b_{ik}=p(O_t=v_k \mid X_t=q_i)$是输出概率；$\pi=\{\pi_i\}$，$\pi_i=p(X_1=q_i)$是初始状态分布。

2. 互信息模型

互信息用来表示两个字之间结合的强度，其公式为

$$\mathrm{MI}(x,y)=\log_2 \frac{p(x,y)}{p(x)p(y)}=\log_2 \frac{p(y|x)}{p(y)}$$

MI 值越大，表示两个字之间的结合越紧密，如“尴尬”、“匍匐”等词。反之，断开的可能性越大。当两个词 x 与 y 关系强时，$MI(x,y)\geqslant 0$；x 与 y 关系弱时，$MI(x,y)\approx 0$；当 $MI(x,y)<0$ 时，x 与 y 称为互补分布。

3. 最大熵模型

熵是用来表示一个随机事件不确定性的量度，即

$$S=-\sum_{x\in\varepsilon}p(x)\log p(x)$$

最大熵模型的作用是在已知条件下选择一个合适的分布来预测可能出现的事件，其主要思想是在只掌握关于未知分布的部分知识时，应选取符合这些知识但熵值最大的概率分布。S 在实验结束前是实验结果不确定性的度量，实验完成后是从实验中获得的信息量。S 越大表明不确定性越大，实验结束后从中得到的信息量也越大。

除了上面的统计算法外，其他的统计和机器学习方法，如决策树、支持向量机 SVM、条件随机域（conditional random fields，CRF）、EM 等也可以用来做分词。另外，针对歧义和未登录词还有一些专门的统计信息，如基于记忆的交叉歧义排除法来统计所有高频伪歧义字段，统计人名、地名用字概率等方法（王思力等，2006）。

2.3.3 规则和统计方法相结合的汉字分词方法

现在多数分词算法都采用规则和统计相结合的方法，这样做既可降低统计对语料库的依赖性，充分利用已有的词法信息，同时又能弥补规则方法的不足。常用的结合方法是利用词典进行初切分，然后用其他的概率统计方法和简单规则消歧进行未登录词识别。例如，ICTCLAS 分词法利用词典匹配进行初切分，得到一个切分词图，然后利用词频信息求词图 N 条最短路径的 N 最短路径法。有些研究者利用词典找出所有的交叉歧义，然后利用 Bigram 语言模型或其变形来消除歧义（王思力等，2006）。

2.4 未登录词及其识别

汉语词语是一个开放集合，时刻处于不断地扩充中，特别是在网络新语言迅速扩张的今天，因此对未登录词的识别就是一个重要的问题。未登录词一般指那些在词典中未能收录且不能通过正则表达式表示的词语，而诸如“整整齐齐”、“亮晶晶”等那些叠词数量众多、构成有规律、能用简单的正则表达式表示出来的不算未登录词。因为数字型合成词的构成方式呈现出规律性，很容易识别出来，很多学者也不将其纳入未登录词的范围。

关于未登录词的分类多种多样，学术界比较认同的分类方法是文献(Chen 和 Bai，1988)中提到的分类方法，对规模为 300 万语料的词汇进行统计，根据其构成方式的不同，它分为以下五个类别。

(1) 缩略语，如“中共”、“非典”等。

(2) 命名识别，又进一步细分为人名、地名、机构名及其他命名实体。

(3) 派生词，由通用词语加上词缀构成，如“机械化”、“先进性”等。

(4) 复合词，由两个或两个以上的核心词语组合而成，是未登录词中数量较多的一部分，如“小汽车”、“校园文化”等。

(5) 数字型合成词，包括日期、时间、电话号码、地址、数字、度量词等，如 1986 年、二千万。

从词性角度来划分，未登录词又分为 14 个小类别(Chen 和 Bai，1988)，其中名词比例最大，占 65.54%；动词次之，占 32.9%；形容词最少，仅占 1.55%。未登录词识别是汉语分词时的一项重要工作，而且是制约分词性能提高的瓶颈之一。

2.4.1　命名实体及其识别

命名实体和新词是未登录词(或称有意义串)的两个重要组成部分，它们能否被正确识别直接影响到分词的性能，国内外研究人员对此关注比较多，并提出很多实用算法。在对命名实体识别的研究中，以人名、地名、音译名识别率较高，正确率和召回率都可以达到 90%以上，而机构名构成规律较为复杂，识别正确率和召回率低一些。识别的方法主要分为基于规则的方法、基于统计的方法和统计与规则相结合的方法。

基于规则的方法主要考虑命名实体的用字规律以及语言环境，分为两类。第一类方法是建立专名资料库，如针对中文姓名识别而建立中文姓名资料库、针对中国地名识别和英美姓名译名识别而分别建立中国地名库和英美姓名译名用字表(孙茂松等，1995；孙茂松和张维杰，1993；宋柔等，1993；沈达阳等，1995)；第二类方法是利用上下文启发信息，限于篇幅不再赘述。

基于统计的识别方法也可分为两类：一类是把机器学习的某种统计模型应用到命名实体识别中，通过对识别过程进行有针对性的建模，将其转化为某种模型的求解问题，常用的模型有决策树模型(Sekine et al.，1998)、隐马尔可夫模型(Bikel et al.，1997)、最大熵模型(Mikheev et al.，1998)、SVM 分类模型(Li，2004)等；另一类是统计汉字的共现概率(王开铸等，1995)或互信息，并以此作为依据判断分词碎片能否构成词。该方法无需引入分门别类的知识库，但该方法提取出来的词有时没有实际意义，如“把这”、“是很”等。

基于规则的方法和基于统计的方法各有利弊：规则方法往往比较严格，存在过学习的问题；统计方法存在数据稀疏问题。将两者有效结合起来能够取得更好的

效果。张华平和刘群(2004)采用统计和规则相结合的方法,采取 Viterbi 算法对切分结果进行角色标注,在角色序列基础上进行字符串识别,在《人民日报》语料上识别准确率达到 95%以上。Wu 和 Jiang(2000)统计常用字在各类命名实体不同位置中(首、中、尾)的可信度,结合上下文可信度的奖惩规则设置经验阈值,对于译名识别的效果比较好。

2.4.2 未登录词与新词识别

语言随着社会的发展而发展,在词汇中的一大表现就是新词语的出现。自古以来,汉语词汇就带有特定时代的烙印,必然会从一个侧面反映出社会政治、经济、文化以及人们价值观念、生活方式的变迁等。例如,在 20 世纪七八十年代,“工分”、“粮票”、“布票”等是人们耳熟能详的名词。随着改革开放的深入,新的词语如雨后春笋般涌现在我们的生活之中,如“政务公开”“价格听证会”“可持续发展”“数码相机”“首付”等,它们真实地反映了社会和经济的飞速发展以及对外交流的日渐频繁。近年来随着互联网的普遍使用和蓬勃发展,大量的新词和网络用语更是层出不穷。

新词虽然也是未在词典中出现的词,但它和未登录词有区别。新词是一个融入了时间的动态概念,而未登录词是相对于词典的概念。未登录词和新词的分类方法不同,它们之间的大致关系如图 2.3 所示。

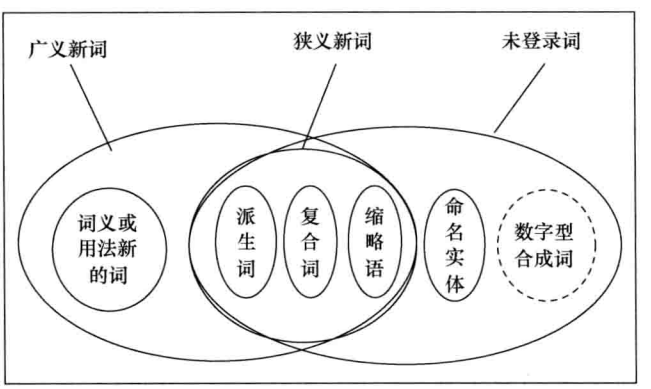


图 2.3 新词与未登录词关系图

从中文自然语言处理的角度来看,广义的新词可以分为三大类。

(1) 命名实体,包括机构名、人名等。

(2) 除命名实体外,词典不曾收录的且具有新词形的词语,如“非典”、“数码相机”等。

(3) 词义或用法发生变化的旧有词汇,但这些新的词义和新的用法,在词典中

没有收入。这类词汇主要可以分为三种：第 1 种是原有词语增加新的意义，如“下课”、“充电”等这些旧有词；第 2 种是原有词语有了新的用法，如以前“火”是名词，现在用作形容词；第 3 种是旧词新用，即不再使用的原有词语又重新启用，如“八卦”等。

在新词发现方法方面，目前主要有基于规则和基于统计两种方法。基于规则的方法其主要思想是根据新词的构词特征或外型特点建立规则库、专业词库或模式库，然后通过规则匹配发现新词。基于统计的方法一般是利用统计策略提取出候选串，然后再利用语言知识排除不是新词语的垃圾串或者计算相关度，寻找相关度最大的字与字的组合。基于规则的方法主要缺点在于局限于某个领域，并且需要建立规则库。基于统计的方法一般都是限于查找较短的新词语，但不便描述词语的内部和外部结构特征，同时对于出现频率很低的新词识别效果不好。

相关工作中，郑家恒和李文花(2002)采用规则的方式识别新词，一方面以新词的构词知识为基础建立新词识别的常用构词库；另一方面从网上词语的特征出发建立特殊构词规则库，并按照规则所取的作用分为“互斥性子串”过滤规则、常规构词规则、特殊构词规则，利用这些规则过滤并确定新词，封闭测试准确率为 91.2%，召回率 95%。规则的方法准确率较高，但总结规则费时费力且受领域限制。由于建立规则的过程需要对新词的特点进行细致的分析，而且规则常与领域相关，不易移植，于是越来越多的研究者开始借助统计方法获取新词信息，并将统计和规则结合识别新词。Chen 和 Jiang(2003)在识别数字、时间、人名等新词时利用了语言学触发规则，如后缀词、姓用字、名用字等。同时，对其他新词，作者计算了一个单字的成词概率，当相邻的两个单字概率的乘积大于某个事先预定的概率时，则认为这两个字组成一个新词。Wu 和 Jiang(2000)是在规则系统中加入了统计信息以提高新词总体的识别效果。贾自艳和史忠植(2004)提出采用基于概率统计技术和规则方法的新词发现方法。针对多字新词抽取精度不高的情况，研究者提出基于质子串分解的新词发现技术(张勇和何婷婷，2005)。对从 Internet 上采集的网页，使用改进的互信息参数抽取结构简单的词汇，在此基础上再用质子串分解的方法将长串分解为更短的子串，然后通过子串之间的关联程度来评估长串成为新语的可能性。邹刚和刘群(2004)对切分后的网页内容，用重复串查找新词语，接着根据给定的时间，建立一个给定时间之前的大规模的词与串的背景词串集合，这个集合里面不仅包括了大部分已有的词语，还包括了噪声和固定搭配，在这个背景词串集合的基础上，通过定义评价函数，对于给定时间以后的词和串进行比较和评价，从中得到新词语候选，最后用过滤规则对候选新词语进行过滤，得到最终的新词语结果。Peng 和 Fang(2004)则完全采用统计的方法，将分词和新词发现作为一个统一的过程，利用 CRF 模型，将词汇特征和领域知识等很好地融入到模型中。同时，发现的新词会加入到词典中，以便能改善分词器的性能，改善后的分词

系统又能更好地识别新词。Meng 和 Yao(2004)提出一种使用 parsing 信息发现新词的方法,在《人民日报》和宾州中文树库的基础上建立了一个基于字的 PCFG 模型,然后在整个句子上建立规则,这种方法充分利用了丰富的上下文信息,克服了传统统计方法的缺陷,可以更有效地识别新词。

2.5 有意义串及其识别

新词和命名实体都是有意义的词语,但有些关键信息,如"建设有中国特色社会主义"、"社会主义新农村"、"高招录取率"、"人民币升值"等,往往不列入词的范围,且随着互联网的不断普及,各种应用场景中这种新短语和新习语大量涌现。这些短语语义明确,往往是反映或描述具体问题和事件的关键信息。对它们的发现和理解具有很大的现实需求。在汉语语言学中,表达语义的关键词语都是名词、动词、形容词和代词,这些词属于开放集合,由于不断有新词产生加入到原有集合中,而副词、助词以及各种虚词只起连接修饰作用,属于封闭集合,几乎不再产生新词。所以,从词性角度来看,有意义串主要是名词、动词、形容词以及名词、动词、形容词为中心词的短语。核心词典中包括常用实词和所有虚词,其中的实词都是有意义串,而副词、介词、连词、助词、叹词和拟声词等虚词在语言中起连接修饰作用,也称为功能词,一般不是有意义串。

为了表征这些关键信息,国内外也有一些学者进行了相关研究,Lee(1997)指出一个有意义的词法模式串是包含任意长度连续字符,它在一组相关文档中是具体的、有意义的。Zhang 等(2000)指出复合词是出现在语料中的稳定字符串,复合词内部的组成成分之间内聚度比较高,而与词语外部两端的词相关性比较弱。Lai 和 Wu(2002)提出 PLU(phase like unit)方法,经过统计方法抽取出来的独立于领域的类似于短语的一个单元。胡吉祥(2006)指出频繁关键模式应该具有强文本表示功能,它具有稳定性、独立性、完整性三个方面的特征。

上述提法都描述了有意义关键信息的某些特征,但是在一定程度上都不够完善。汉语词语的定义本来就相对模糊,而新词又不好界定,没有统一标准。所以,应从现实需求出发,突破词语概念的界限,将包含具体语义,并且能够作为一个灵活独立的语言单元在不同语境中使用的字符串称为有意义串。具体来说,有意义串应该满足以下几个特性。

(1) 语义上:表意完整单一,所指明确,在意义上有一定的完整性和专指性。但借助计算机来做语义理解比较困难。我们发现有意义串在结构、语用、统计上还有一些其他特性,这些性质能够表示成计算机可计算的形式,我们将通过计算它们来辅助判断一个串的语义完整性。

(2) 结构上:结构较稳定,具有一定的凝固性。

(3) 语用上:是一个独立的语言单元,使用环境灵活多样。

(4) 统计上:在大规模真实文本中具有一定的流通度,并非临时性的组合,可重用性强,具有统计上的意义。

“市场经济”、“八荣八耻”、“科学发展观”等串都满足上述四个特性,是有意义串的典型例子。而“你的”、“还是这样”虽然在语言中出现频繁,但没有包含具体的语义,只是模糊的广泛的表述,不属于有意义串;“邓小平理”作为“邓小平理论”的子串表达语义不够完整,也不是有意义串。句子虽然是串,但其语义复杂,有一定的逻辑关系,不具有灵活性,不属于有意义串研究的范围。有意义串与语言学中的组块 chunk 概念也有所不同,组块是符合一定句法功能的非递归短语,由核心词和从属成分构成,而有意义串关键强调语义功能,可以是词或者短语。例如,“拿到”在句子中是一个组块,但不是有意义串,语义不够完整。

2.6　词典组织与管理

在进行自然语言处理(如分词、主题词抽取)时,一般需要词典的支持。为了提高处理效率,需要关注词典的组织。目前用于组织词典的索引方法包括基于 Trie 索引树的方法和基于散列哈希表的方法等。

2.6.1　基于 Trie 索引树的词典管理

基于 Trie 索引树的词典机制一般是对词的首字计算其哈希值,建立首字哈希表,然后建立 Trie 索引树,如图 2.4 所示。

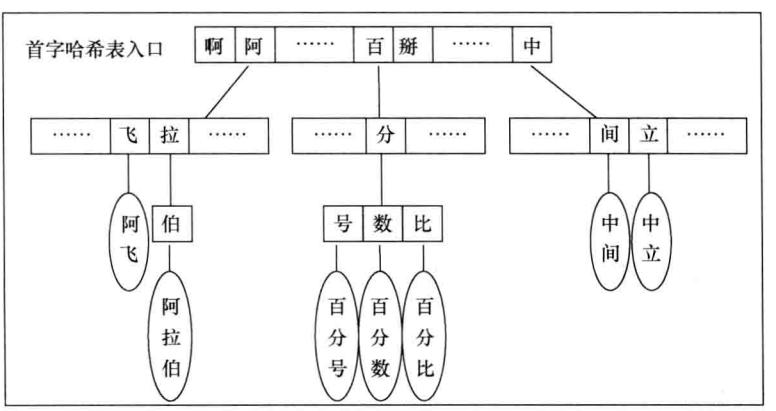


图 2.4　基于 Trie 树的词典结构

Trie 索引树的优点在于检索时只需沿树链进行逐字匹配,不需预知待查词的长度。查找一个词所需的时间只与该词的长度有关,其时间复杂度为 $O(n)$(其中

n 为被查询词长度)。但是树的构造和维护比较繁琐,且由于单词树枝比较多,所以 Trie 索引树的空间浪费比较严重。李庆虎等(2002)的实验结果表明,Trie 树结构的词典占用空间要比基于单纯哈希索引的词典占用空间大近一倍。

2.6.2 基于哈希表的词典管理

这种方法是在表项的存储位置和关键码之间建立一个确定的函数对应关系 Hash(),使得每关键码与该结构中唯一的存储位置相对应。基于哈希索引的词典机制就是构造一种哈希函数来计算词语的哈希值,将哈希值相同的词语放入一个通常称为"桶"的集合内。检索时,先计算待查词的哈希值,然后进入相应的哈希桶内进行二分查找,常用的查找方法有首字哈希和两字哈希法。首字哈希是利用首字生成哈希值,将哈希值相同的词放在同一个哈希桶中,然后在桶中进行二分查找。参照文献(孙茂松等,2000),图 2.5 给出基于首字哈希的二分查找词典结构。

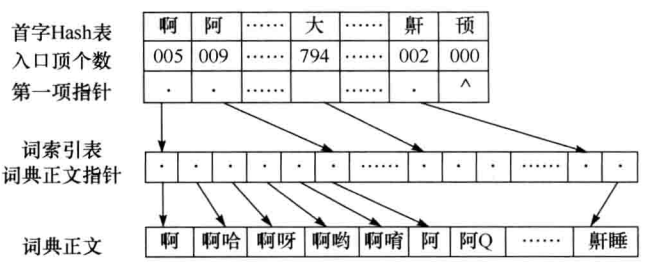


图 2.5 基于首字哈希的二分查找词典结构

两字哈希是在首字哈希基础上的一种改进。目前主要有两种方法:张华平(2005)提出的 ICTCLAS 系统中采用的根据前两字内码计算哈希值,建立一级哈希索引表,然后进入哈希值相同的表中进行二分查找,这种方法同首字哈希词典结构(只不过它是用前两个字做哈希运算);第二种方法是对词语的前两个字依次顺次建立哈希索引,构成深度为 2 的 Trie 子树,对剩余的字符串再进入相应的哈希桶内进行二分查找(李庆虎等,2002)。参考相关文献(李庆虎等,2002),图 2.6 给出了双字哈希词典的结构。

基于哈希表的词典性能与哈希函数的好坏有很大的关系。一个好的哈希函数要能够使地址分布尽可能均匀,或者尽量把出现频率高的词语映像到稀疏的空间,出现频率低的映像到稠密的空间,以减少处理冲突的时间。可以说,采用哈希索引机制的词典查找时间复杂度为 $O(\log_2 N)$(N 为哈希值相同的词条数最大值)。双字哈希算法的提出是利用中文词语大多数是二字词的特点。据统计,一个容量为 42 425 条的词典中单字词 1650 条、双字词 30 770 条,一共占了所有词条的 76.42%(李庆虎等,2002)。在文档中,通常多字词出现的频率也比双字和单字词

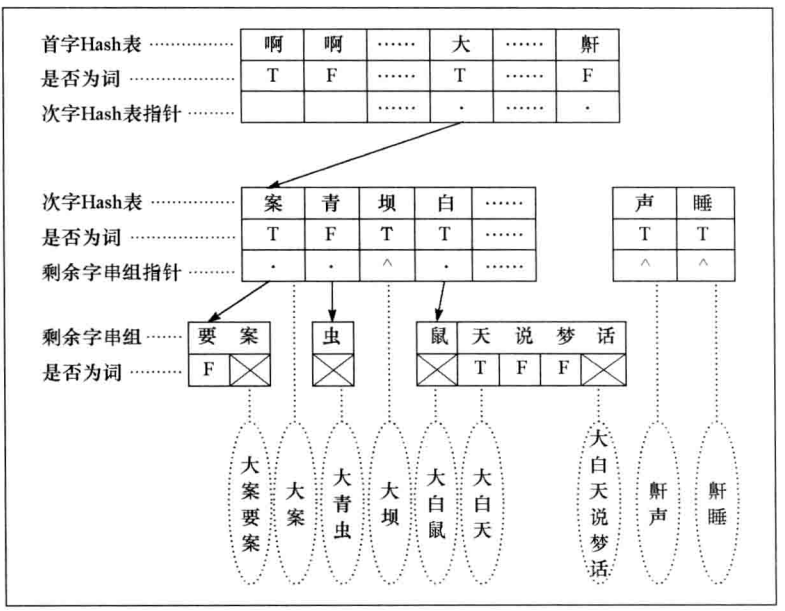


图 2.6　双字 Hash 词典结构

要低，所以双字哈希方法的查找和切分速度比首字哈希有提高。因此，李庆虎等(2002)的实验在词典查询和匹配切分速度上，采用双字哈希结合逐字二分的算法最快，Trie 树算法略低，但差距不明显。考虑到算法实现上的影响，可以忽略这个差别，认为二者速度差不多，但首字哈希是最慢的；在空间占用上，Trie 树明显高于其他几种算法。综合起来看，双字哈希结合逐字二分的方法在速度上和 Trie 树差不多，但在空间耗费和结构维护上要优于 Trie 树结构。

2.7　文本分类

文本自动分类的发展大致经历三个阶段：第一阶段(20 世纪 60 年代早期)主要进行自动分类的可行性研究，代表人物是 Maron 和 Borko，前者提出用标引词代替文本，根据标引词在类中出现的概率来确定类别；后者提出将多个标引词用少数因子来代表，从而确定最终类别的因子分析法。第二阶段(20 世纪 60 年代中期～70年代中期)进行自动分类的实验研究，这一阶段的主要特点是探索各种可用的分类方法。在这之后的第三阶段使自动分类开始实用化，并在邮件分类、信息过滤等方面取得应用，如麻省理工学院为美国白宫开发的邮件分类系统、卡内基集团为路透社开发的 Construe 系统等。我国的文本分类研究始于 20 世纪 80 年代，大体经历了可行性探讨、辅助分类系统、自动分类系统三个阶段。总的来说，中文文

本分类还处于实验研究阶段，正确分类率约 60%～80%（苗夺谦，2007）。在 Reuters-21578 路透社新闻数据集中，经对几个常用算法性能的实验表明，在召回率平均保持在 76%～81%的情况下，准确率平均只有 82%～91%（Yang，1999a，1999b）。

相关方法中，词匹配法定义一个分类列表，每个分类包含一系列的属于这个类别的词，在分类文档时看文档中的词多数出现在哪个分类里（或者属于哪几个分类），就可以判定其所属类别。这种方法比较简单，要求某个类别的词汇表要非常全面，但当遇到一个包含新词的文档时，即使这个词的出现频率非常高，也不会将其划分到正确的类别，由于网络新词汇的急剧涌现，也使得这种方法难以保证分类的准确性；基于知识工程的分类法需要编制完整的推理规则，这种方式严重依赖推理规则的质量，在对表达方式多样的文本进行处理时，其适应性是得不到保证的（Feolman，2007）。统计学习法是找出一些与待分类的文档属同一类别的文档当成样本并由人工进行分类，分析这些已经分好类的文档，从中找出关键词与类别间的联系，用这些学习得到的知识指导对后续文档的分类工作。随着网络海量信息的涌现，以及各种未登录新词的剧增，使得多数传统方法无法满足当前需要。目前，基于机器学习和统计学习的自动分类方法逐步取代了基于人工和词典分词的传统分类方法。

在文本分类算法中，朴素贝叶斯是一种简单的线性分类器。它在文本分类中应用得非常普遍，并且经常被用来作为分类器比较时的 Baseline，其决策规则如下，即

$$C=\arg\max_{c_k}(p(c_k|d_i))=\operatorname{argmax}c_k\left(\frac{p(c_k)p(d_i|c_k)}{p(d_i)}\right)$$

其中，c_k表示第 k 个类别；d_i表示编号为 i 的文档。

贝叶斯分类法及在此基础衍生出的三种分类方法，即基于最小风险 Bayes 决策、尼曼-皮尔松分类器、最小最大决策，要求已知先验概率及条件概率密度函数，但类别总体概率分布和各类样本的概率分布函数往往是未知的，这就限制了其应用。基于支持向量机的分类算法适用于小样本的有指导分类，在理论上还有待进一步完善。KNN 算法对待分类样本要计算它到全体已知样本的距离，之后才能求得 K 个最近邻点，计算量较大，因此常事先对已知样本点进行剪辑以便去除对分类作用不大的样本，或者是用空间换时间，事先将所有样本点的两两距离计算出来以备检索，前者易产生新的误差，后者将占用过多的存储空间。向量空间模型依据语料库中训练样本和分类法建立类别向量空间，并用类别向量空间中的各个类别向量与文本特征向量进行相似度比较，用相似度最大的类别作为待分类文本的类别，但它对样本选择和类别体系要求严格，一般来说偏重于专业领域。决策树分类法需要依据不同的应用来合理确定分类树结构，其不足之处是要用较多的时间

建立分类器，另外对不同的层次使用的特征和分类方法进行选择以达到最优，也是一项较困难的工作(尹中航，2002)。最小方差映射法较适用于用户需求相对固定的场合，但当用户数据发生变化时适应性不强。基于神经网络分类算法适用于样本规模相对较少且用户需求相对稳定的场合，但可能存在过学习问题，且较多依赖人的经验。基于模糊数学的分类法可以与各种已有技术结合使用，但在一定程度上可能引入主观因素，有时分类效果并不好。有的算法还存在主题漂移问题(Forman，2006；Hand，2006)。研究表明，在 Reuters-RCV1 数据集上，支持向量机算法优于 KNN 和 Rocchio (Lewis，2004)，Joachims(2006)提出一种比朴素贝叶斯分类算法更优的具有线性时间复杂度的算法。呼声波和刘希玉(2008)对常用的集中文档特征提取方法进行了比较研究，提出一种改进的 CHI 算法。

2.8　文本聚类

聚类分析是一种无指导的机器学习方法，在机器学习、统计分析、模式识别、数据挖掘、生物学等许多领域得到了广泛的研究与应用。聚类的基本目的是将数据对象按照一定的标准分成若干个簇，使得同一个簇中的对象之间相似度较大，不同簇之间的对象相似度较小。文档的聚类分析与一般的聚类分析类似，往往包括如下步骤(Jain and Pubes，1988)。

(1) 文本表示，往往包括特征抽取和特征选择，把数据对象表示成适合算法可计算的形式。

(2) 相似度度量根据领域知识定义模式之间的距离测度公式。

(3) 聚类或者分组算法体系。

(4) 数据抽象表达(如果需要)。

(5) 评价输出结果(如果需要)。

这里主要对前三个步骤进行介绍。

2.8.1　文本表示

文本聚类分析首先要考虑的是文本表示问题，即如何从一个电子文本的符号和文字中抽取出特征，通过这些特征来表示相应的文本，利用这些特征数据进行聚类分析。不同的文本表示形式可以选择不同的表示模型，如向量空间模型、概率模型、语言模型等，可以选择不同的特征选择方法，利用 LSI 等不同的降维策略。

常用的表示方法是向量表示法。向量空间模型将文档表示成一个向量，向量的每一维表示一个特征，该特征可以是一个字、一个词、一个 n-gram 或某个复杂的结构。通常情况下，利用向量空间模型表示文档时，需要对于文档进行切分(即前述的中文分词，对英文而言是通过词的分界符识别单词)、停用词处理、英文词的词

形还原或者提取词干，经过这些步骤的处理，基本上可以得到一系列词并将其作为文档的特征。所有的这些词构成一个空间，每个词对应空间中的一维。每个文档可以用文档中的词来表示，这些词及其权重构成一个向量。向量的每个维度描述对象的一个特征，每个特征的重要度通常采用 TF * IDF 的计算方法来衡量，更多的工作是通过参数调节来获取更好的效果。在著名的信息检索系统 Smart 中提出过一套词权重计算的命名，该命名综合了 TF * IDF 中的多种变化，把词的权重计算归结为三个组成部分，即词频 TF、反向文档频率 IDF 和基于文档长度的规格化因子。每个部分都有不同的计算方法，这些不同的变化构成了文档的不同表示方式。常用的表示公式为

$$w(t,d) = \frac{1+\log_2(\mathrm{tf}(t,d)) * \log_2(N/n_t)}{|d|}$$

其中，$w(t,d)$表示词 t 在文档 d 中的权重；$\mathrm{tf}(t,d)$表示词在文档 d 中的频率；N 表示整个语料中的所有文档数目；$|d|$表示文档向量的长度。

有的学者尝试用不同的方法来完成相关工作。例如，考虑基于概念的聚类（Hotho and Stamme，2002；卜东波等，2002），而不是基于简单的词的聚类；考虑利用文档中每个词的互信息向量（Pantel and De kang，2002）来聚类，而不是权重向量等。此外，也有从效率方面来考虑，先用一字词聚类，再用二字词进一步聚类提高聚类精度（李振星等，2003）。

2.8.2 相似度度量

在文本聚类中，除了将文本进行某种表示外，还必须对文本进行相似度度量。不同的度量方法会出现不同的聚类结果。文本聚类主要区分三个相似度度量，即

（1）文本与文本之间的相似度度量。

（2）文本簇与文本簇之间的相似度度量。

（3）文本与文本簇之间的相似度度量。

限于篇幅，这里只介绍前两者的相似度度量方法。

1. 文本与文本之间的相似度度量

在向量空间模型中，文本与文本之间的相似度往往采用余弦距离。在语言模型中，文档与文档之间的距离可以采用 KL 距离来计算，即

$$\mathrm{Score}(D,Q) = \sum_W p(w \mid \theta_Q)\log\frac{p(w \mid \theta_D)}{p(w \mid \theta_Q)}$$

对于以向量表示的文档，除了表示模型本身的距离度量方法之外，在传统的聚类分析中还存在多种其他距离度量方法，如明考斯基距离，即

$$d_{ij} = \left[\sum_{k=1}^{n} | x_{ik} - x_{jk} |^r\right]^{1/r}$$

2. 文本簇与文本簇之间的相似度度量

在聚类分析中，有时需要衡量文档簇和文档簇之间的相似度，从而选择簇与簇之间的合并或者拆分。文档簇与文档簇之间的相似度也存在多种定义，常见的有最小距离、最大距离、平均距离、质心法、离差平方和、代表点等。下面列出常见簇 C_1 和 C_2 之间相似度的度量方法。

1）最小距离

使用两类间最近两点的距离来描述两类间的相似程度，即

$$D(C_1, C_2) = \min\{D(X,Y) \mid X \in C_1, Y \in C_2\}$$

2）最大距离

使用两类间最远两点的距离来描述两类间的相似程度，即

$$D(C_1, C_2) = \max\{D(X,Y) \mid X \in C_1, Y \in C_2\}$$

3）平均距离

使用两类间两两样本点距离的平均值来描述两类间的相似程度，即

$$D(C_1, C_2) = \frac{1}{| C_1 | * | C_2 |} \sum_{X \in C_1} \sum_{Y \in C_2} D(X,Y)$$

4）质心法

使用两类质心之间的距离来描述类间相似性，即

$$D(C_1, C_2) = D(\overline{X}, \overline{Y})$$

5）离差平方和法

使用簇内的方差来度量。正确的聚类应当使得簇内方差尽量小，而两簇方差尽量大。簇内方差定义为

$$S(C) = \sum_{X \in C} (X - \overline{X})^{\mathrm{T}} * (X - \overline{X})$$

因此，两个簇间的距离为

$$D(C_1, C_2) = S(C_1 \cup C_2) - S(C_1) - S(C_2)$$

2.8.3　聚类算法体系

聚类算法体系的划分可以基于不同的标准，如凝聚还是分裂、增量还是非增量、确定还是随机、Hard 还是 Fuzzy 等，因此聚类算法体系的划分在各文献中并不统一。这里采用 Han 和 Micheline(2001)的划分方法，把聚类算法分为层次聚类、划分聚类、基于网格的聚类、基于密度的聚类、基于模型的聚类。

1. 基于层次的聚类

层次聚类将文本对象组织成树，每个叶子节点是一个簇，该树称为一个聚类谱系图。根据树是按自底向上还是自顶向下方式而生成的，可以划分为凝聚式和分裂式层次聚类。层次聚类算法复杂度高，可根据需要获得不同粒度的类(Han and Micheline，2001)。目前在层次聚类研究中出现了一些结合其他聚类方法的多阶段聚类分析技术，比较有代表性的有 BIRCH(balanced iterative reducing and clustering using hierarchy)、CURE(clustering using representatives)、Chameleon 等都是基于层次思想的组合算法。BIRCH 是一个增量式动态聚类分析算法，通过数据结构 CF(clustering feature)树对数据进行聚类。Chameleon 是一个两阶段的聚类分析算法。第一阶段是构建相似度矩阵，从矩阵中再构建 K 最近邻图，然后对图进行划分得到小的簇；第二阶段是对小簇进行动态和并，合并时采用互连性与近似度来度量簇与簇之间的相似性。

2. 基于划分的聚类

基于划分的聚类一般不需要建立起类之间的层次结构，只需要将数据集进行划分，形成一个平面的类结构。比较典型的方法如下。

1) K-Means 聚类

K-Means 聚类分析算法是一种简单高效的聚类分析算法。K-Means 将聚类对象划分成为 K 个簇(K 是事先确定的参数)，每个簇用一个中心表示，该中心定义为簇中对象的平均值。K-Means 聚类的缺点是必须事先给出 K、对数据输入顺序敏感、不适于发现很大的簇、对噪声和孤立点敏感等。在文本聚类中如果不进行特征选择，特征维度过高的话，很容易扎堆成几个大类和众多小类。

2) K-Medoids 聚类

K-Medoids 聚类是与 K-Means 类似的一种聚类分析方法，其主要区别在于 K-Means 使用所有点的均值来代表簇，而 K-Medoids 则采用类中某个数据对象来代表簇。PAM(partitioning around medoid)是最早提出的 K-Medoids 算法之一。与标准的 K-Means 相比，PAM 通过反复计算随机选择的点与原有 K 个中心的交换代价来决定如何形成新的中心点。PAM 算法复杂度高，因此由 PAM 演化出了 CLARA 以及 CLARANS 用于对大规模空间数据的聚类分析。CLARA 从原始对象随机采样得到 K 个中心点，可以进行多次采样得到更优的 K 个中心。CLARANS 比 CLARA 更进一步，每一次应用 PAM 时，面对的不是一个固定的采样样本。如果对于 K-Medoids 的簇中心重新定义，例如采用类似 CURE 的代表点的方式来代表一个簇，可以演化出新的算法。

3）最近邻聚类

最近邻聚类是一种简单的聚类分析。该算法认为，如果某个待处理的对象与现有的所有簇的最小距离小于给定阈值，那么将该对象归入该簇，否则将该对象作为一个新的簇。最近邻聚类不需要事先决定结果类数，一般适合类内距离小、类间距离大的情况，结果与样本的输入顺序有关，也与阈值 T 有关。随着样本集的增大，产生的结果类数可能越来越多，处理的速度会越来越慢。

3. 基于密度的聚类

基于密度的聚类算法将簇看成是数据空间中被低密度区域分割开的高密度区域(Han and Micheline，2001)。密度定义为单位体积内的点数，簇的内部密度大。由于密度是一个局部概念，这类算法又称为局部聚类；基于密度的聚类算法通常只扫描一遍数据库，所以又称为单遍扫描聚类。基于密度的聚类中比较典型的算法有 DBScan(desity-based clustering)、OPTICS(ordering points to identify the clustering structure)、DENCLUE(density-based clustering)。

对于空间中的一个对象，如果在给定半径 Eps 的邻域中的对象个数大于给定的数值 Minpts，该对象称为核心对象，否则称为边界对象。由一个核心对象可达的所有对象构成一个簇。DBSCAN 在研究中得到比较多的关注，GDBSCAN(generalized DBSCAN)、DBCLASD(distribution based clustering of large spatial database)、FDC(fast density-based clustering)都是对 DBSCAN 的进一步扩展和优化。

4. 基于网格的聚类

基于网格的聚类首先把数据空间划分为一定数目的单元，以后所有的操作都针对这些单元进行，通过对单元的操作产生结果簇。这种方法的优点是处理速度快，处理时间独立于数据对象的数目，仅依靠网格空间的维度。基于网格的代表性例子有 STING(利用存储在网格中的统计信息)、WaveCluster(利用小波变换来聚类对象)、CLIQUE 等(Han and Micheline，2001)。

5. 基于模型的聚类

基于模型的聚类方法为每一个簇假定了一个模型，寻找数据对给定模型的最佳拟合。基于模型的方法有统计学方法 COBWEB 和自组织映射(SOM)神经网络方法等。其中自组织特征映射方法首先对文档降维，把特征维度降低到足够小，然后进行 SOM 聚类。该算法的意图是把 SOM 算法扩展到处理高维数据对象的领域，适合用户对文档内容所属领域不太了解的情况。

与普通的聚类场景相比，文本聚类属于高维空间上的聚类分析。高维空间的数据经常呈现奇怪的特征，如噪声的比例增加、网格单元的个数呈指数级增加、数

据空间的稀疏性大、聚类的密度低等，这在数据挖掘中称为维数灾难(Han and Micheline,2001)。许多聚类分析的算法，如 DBSCAN、STING、Wave-Cluster 等，随着维数的增加，算法的空间和时间的复杂性往往增长得很快，所以很难实用化。对于传统的聚类算法来说，当其应用在文本聚类的场景中时，除了要调整好参数以外，还要尽量避免高维特征带来的影响。

2.9 话题识别与跟踪

在信息爆炸的时代下，一方面网络信息的规模急剧膨胀，另一方面信息又凌乱无序，有价值的信息常常湮没在大量冗余信息中，对其发现和管理变得越来越困难。为此，以话题为主线对信息进行组织引起了越来越多人的兴趣，但其困难也是显而易见的。同一个话题，往往存在着多个角度、多个侧面的描述与分析，这些信息通常孤立地分散在不同的角落；话题本身也会随着时间而不断地发展变化，描述它的信息会以不同的面目出现在不同的时间。如何对分散多变的话题信息进行有效的组织就成为需要考虑的问题。

话题识别与跟踪就是在这种情况下应运而生的。通过话题识别与跟踪，人们可以将关于话题的分散信息有效地汇集并组织起来，从而帮助用户发现话题中各种因素之间可能隐藏着的关系，从整体上了解一个话题的全部细节及该话题与其他话题的联系。

话题识别是一个无监督的机器学习过程，话题识别也是围绕文本聚类的相关技术来展开的，通过调整算法使之更加适合对话题的处理。话题识别与跟踪的概念最早产生于 1996 年，当时美国国防高级研究计划委员会提出要开发一种新技术，能在没有人工干预的情况下，自动判断新闻数据流的主题。早在 1997 年，卡内基梅隆大学、Dragon 系统公司、BBN 公司以及马萨诸塞大学的研究者对这一个新的问题领域进行了初始的研究，主要是定义话题识别与跟踪研究的内容和开发相关的初步技术。这些初始研究的目的是要确定来自信息检索领域的基于主题的技术在多大程度上能够用来解决基于事件的信息组织问题。话题识别与跟踪评测采用的语料是由语言数据联盟提供的 TDT 系列语料，目前已公开的语料包括 TDT-Pilot、TDT2、TDT3、TDT4 和 TDT5，这些语料都人工标注了若干话题作为标准答案。通常在每次评测中，先前开发的语料被用做训练集，而新标注的语料作为测试集。总的来看，TDT 系列评测会议呈现两大趋势：一是努力提高信息来源的广泛性，不仅包括互联网上的文本数据，还包括来自广播、电视的语音数据；二是强调多语言的特性，从 1999 年开始，TDT 会议引入了对汉语话题的评测，2002 年又增加了对阿拉伯语的测试。

在话题识别任务的定义中，系统必须增量地处理话题的信息，在某一时刻只能

看到有限的时间窗口内的报道(Wayne,2000)。卡内基梅隆大学采用了向量空间模型,在时间窗口内,利用组平均聚类的方法先把报道聚成小类,再利用最近邻聚类将小类与已有的信息合并。组平均聚类属于层次凝聚式聚类的变种。实验证明,这种按照时序进行组内优先聚类的方法是比较合适的,因为它把时间相近的报道聚到一起。实验发现,组平均聚类更适合突发性的事件,而最近邻聚类适合持续性事件,把二者混合起来能达到比较好的效果(Yang,1999b)。与之相似的是 IBM 采用的两层聚类方法,当时间窗口增大时,系统的效果提高得比较明显(Franz et al.,1999)。马萨诸塞大学采用向量空间模型和语言模型。向量空间模型中尝试了中心质点法和 K 最近邻法;语言模型中计算先前的哪个话题 M 最有可能产生新到达的文档 D,即 $P(D|M)/P(D)$,使用一元模型,$P(D \mid M) = \prod_i P(d_i \mid M)$,$d_i$ 表示 D 中的符号,并采用最大似然估计来计算 $P(d_i \mid M)$。实验发现,采用 K 最近邻聚类,$K=1$ 时效果是最好的。与此同时,Dragon 系统公司和 TNO(荷兰国家应用科学研究院)采用的是语言模型。其中,TNO 在最近邻聚类中引入了聚类结果调整的策略,即当一个时间窗口内的报道处理完毕时,重新比较每一篇文档和已有的类,把文档调整到最相近的类中,并且如果一个类结果在一段时间内不发生变化,这个类就固定了。实验表明,结果调整的策略提高了系统的性能。

话题识别处理的对象是热点话题的新闻报道。因此,研究者采用了符合话题特点的文本表示方法,主要是挖掘合适的话题特征词和合理地计算特征词的权重。在话题识别中,一个明显的现象是事件发生的时间、地点、人物、机构等命名实体对话题的区分作用比较大。因此,一般的研究者都对命名实体进行了有针对性的处理(Yang et al.,2002)。

对于一个新闻报道的来源来说,可能有很多特有的惯用词汇。例如,对于 CNN 网站,词汇"CNN"就可能频繁出现,而恰恰就是这样的词汇,对该网站来说表达力比较弱。因此,研究者针对每一个新闻来源内的文档单独进行词频的统计(Franz et al.,1999)。马萨诸塞大学比较了不同的权重计算方法,包括 Inquery TF_IDF、TF_{only}和 TF_IDF 等。跟话题识别相关的特征策略还包括采用增量的文档倒排频率(IDF)计算模型、过滤短文档、考虑词在文档中的位置(即报道首段的词权重较大)等。

尽管有些技术不是应用在话题识别的任务中,但其特征的调整策略对话题识别也适用。相关工作中,Carthy(2002)利用 WordNet 构建词汇链(即文本中相关词汇的信息串),以此作为文本表示的主要特征。Makkonen 等(2004)提出了事件向量的概念,并采用基于本体的方法进行地名相似度的计算。Brants 等(2003)采用基于倒排事件频率的特征权重计算方法,即一个词在各个事件中出现的次数越多,这个词就越倾向于不重要。David(2003)利用时间词与地名词的绑定来对事件

进行识别。

2.10 句子及其检索

如果将句子看成是小型文档，可采用基于文档的信息处理办法并加以适当改进来完成句子级的检索，可将句子检索方法划分为传统的文档检索方法、信息过滤方法、分类方法、语义比较方法、隐马尔可夫模型方法与自动文摘方法等。

2.10.1 传统的文档检索方法

单个句子本身可以看成是一个微型文档。研究者普遍采用的是传统的文档检索方法，常用方法主要有经典的 TFIDF 方法、语言模型方法等。给定查询 q 与句子 s，经典的 TFIDF 可以采用下面的公式计算出该句与查询的相关度，即

$$\text{score}(s) = \sum_{t \in q} \log(\text{tf}_{t,q} + 1) \log(\text{tf}_{t,s} + 1) \log\left(\frac{n+1}{0.5 + \text{sf}_s}\right)$$

其中，tf 为词语 t 在句子中出现的频率；sf 为出现 t 的句子数目。

在语言模型中，往往采用 KL 距离的方法计算句子 s 与查询的相关度，即

$$\text{score}(s) = \text{KLD}(q \mid\mid s) = \sum_{w \in V} p(w \mid q) \log \frac{p(w \mid q)}{p(w \mid s)}$$

实际上，句子和文档仍存较大差别，不能简单地照搬文档检索的方法，还需针对句子特点进行改进。为此，研究者引入了各种查询扩展、文档扩展以及参数调整策略。

2.10.2 信息过滤方法

信息过滤方法的基本思想是将主题作为初始的用户兴趣背景，将句子看成是文档集，过滤条件为句子与用户兴趣的相关度，而过滤条件采用的阈值是在训练集上通过适当的机器学习算法得到，目标是使综合指标 F 值最大。与文档信息过滤不一样的是，句子信息量往往太少，因此一般需要做进一步扩展。明治大学的研究者在信息过滤的实现过程中，认为用户兴趣主要来源于主题的简介部分，同时还引入了概念模糊集对用户兴趣和句子进行概念扩展，其中 X 为输入，C_1，C_2，…，C_n 为概念集合中所有的概念（图 2.7）。在分别计算它们之间的相似度后，乘以不同的加权系数 a_{ij}，最后得到扩展之后的输出概念。

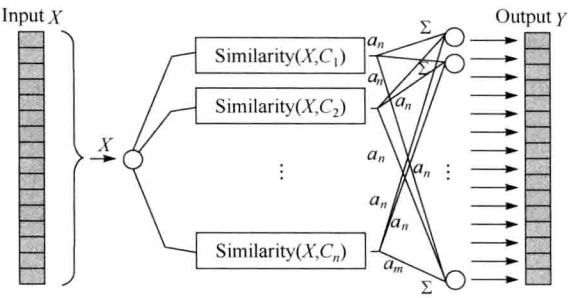


图 2.7　引入概念模糊集扩展信息过滤输入的示意图

2.10.3　分类方法

使用分类的句子级检索方法，主要根据词语的分类特征属性来判断所在句子与主题的信息相关度。Taoufiq 和 Mothe(2004)依据主题分析的结果定义了四种类型的词汇，即高相关词(highly relevant terms，HT)、低相关词(low relevant terms，LT)、不相关词(non-relevant terms，iT)与逆相关词(Inverse relevant terms，IT)。其中，逆相关词是从主题详细描述中不相关文档的叙述文字中抽取，而其他类别的词均从剩余部分中抽取，相应权重根据其出现频率和在主题中的位置来综合评估，最后根据设置的阈值和权重的大小，得到不同类别的词汇集合。给定一个主题 Q_k 和句子 S_j，句子 S_j 与主题 Q_k 相关的充要条件为

$$\text{score}(S_j,Q_k)>f\left(\frac{|\text{LS}_j|}{|\text{LS}_j|+|\text{HS}_j|}\right)\cdot|\text{HT}_k|+g\left(\frac{|\text{HS}_j|}{|\text{LS}_j|+|\text{HS}_j|}\right)\cdot|\text{LT}_k|+\alpha$$

其中

$$\text{score}(S_j,Q_k)=\sum(\text{weight}(t_i,S_k)\cdot\text{weight}(t_i,Q_k))$$

LT_k 为主题 Q_k 中的低相关词集合；LS_j 为同时出现在句子 S_j 和主题 Q_k 中的低相关词集合；HS_j 为同时出现在句子 S_j 和主题 Q_k 中的高相关词集合；$|X|$ 表示的是集合 X 中元素的个数；$\text{weight}(t_i,S_j)$ 为词语 t_i 在句子 S_j 中出现的频率，$\text{weight}(t_i,Q_k)$ 由词语 t_i 在主题 Q_k 中出现的频率及其位置综合加权得到。

除此之外，还有另外一种基于文本分类思想的句子检索方法，即将相关和不相关视为两种不同的类别，分别采取已有的数据进行训练，最终得到一个句子检索的分类器。NTT 通讯公司尝试了 SVM 的方法(Hideto et al.，2002)，也有研究者利用给定的部分结果训练分类器，限于篇幅，不再赘述。

2.10.4　语义比较方法

语义比较方法主要是借助于现有的语义体系来实现不同词语之间的相似性计

算，然后综合计算得到句子与查询之间的相关度。对于句子来说，按照词形进行精确匹配往往过于严格，无法发现句子与主题之间的潜在关系。Tsai 和 Chen(2002)提出了一种比较简单的语义比较的句子检索方法。首先对于两个不同形的词语 w_1 与 w_2，将其对应的概念在 WordNet 层次结构中的最短路径长度定义为两个词的语义距离 $\mathrm{dist}(w_1, w_2)$(如图 2.8 中“sky”和“universe”的语义距离为 4)。设定一个阈值，如果语义距离小于该值，则两个词相似，否则两者没有关系。

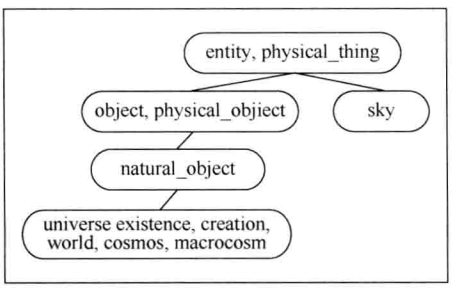


图 2.8 语义距离示意图

对于任意两个句子，如果一个句子中的名词或动词和另外一个句子精确匹配，则相应的匹配度加 1；如果不存在精确匹配，则计算两个词语之间的语义距离 $\mathrm{dist}(w_1, w_2)$，如果 $\mathrm{dist}(w_1, w_2)<4$，则相应的匹配度加 0.5；每个词语只能匹配一次。句子之间的相似度由名词相似度和动词相似度来表示，即

$$\mathrm{sim}(s_1, s_2)=\mathrm{noun_sim}(s_1, s_2)+\mathrm{verb_sim}(s_1, s_2)$$

$$\mathrm{noun_sim}(s_1, s_2)=\frac{m}{\sqrt{ab}}$$

$$\mathrm{verb_sim}(s_1, s_2)=\frac{n}{cd}$$

其中，s_1 和 s_2 分别表示两个句子；m 和 n 分别表示名词和动词匹配的值；a 和 b 分别表示 s_1 和 s_2 中包含的名词数；c 和 d 分别表示 s_1 和 s_2 中包含的动词数。

2.10.5 隐马尔可夫模型方法

与简单贝叶斯方法相比，隐马尔可夫模型独立性的假设更少，即它不会假设第 i 个句子的相关概率与第 $i-1$ 个句子是否相关或完全无关。马里兰大学的研究者借鉴自动文摘的做法提出了一种基于隐马尔可夫模型的句子检索方法(Conroy, 2004；Conroy et al. ,2003)，观测的是和句子中词语相关的特征。

(1) 句子中标志词(指那些相对于通用语言环境来说更能代表当前主题且在当前主题中更可能出现的词语，可采取互信息等统计方法发现)的数目 n_{sig}，其观测值为 $o_1(i)=\log(n_{\mathrm{sig}}+1)$。

(2) 句子中主题词(subject tokens)的数目 n_{sub}，其观测值为 $o_2(i)=\log(n_{sub}+1)$；主题词属于标志词的一个子集，指的是那些出现在标题或者头条中的标志词。

(3) 句子中的词数(tokens)n_{tok}，其观测值为 $o_3(i)=\log(n_{tok}+1)$。

(4) 句子在当前文档中的位置。

隐马尔可夫模型共有 $2s+1$ 个状态(s 个相关状态，$s+1$ 个不相关状态)，图 2.9 给出的是 7 个状态的马尔可夫链。

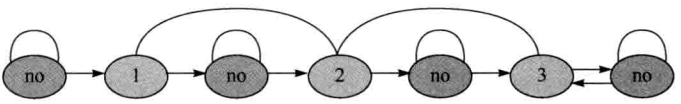


图 2.9　7 个状态的马尔可夫链(其中 4 个为不相关状态)

状态之间的转移概率在训练集上采用最大期望估计的方法学习得到，最后形成马尔可夫过程的状态转移矩阵。每个状态上的输出概率 $b_i(O)=P(O|\text{状态 } i)$，其中 O 为观测到的特征向量。为简化计算，各特征间假定相互独立，直接采用联合概率估计。

最后，句子检索的过程就转化为搜索一条概率最大的路径，路径中的节点对应的就是检索出来的相关句子结果集。

2.10.6　自动文摘方法

新信息检测最早是在自动文摘中提出的，自动文摘的目标是提取能够概括当前文档、没有冗余的句子集合，这与新信息检测是一致的。人们想到可将一些成熟的自动文摘方法应用于新信息检测，不过还需要进一步依据句子的特点进行改进。Michigan 大学的研究者一直尝试将研制的多文档自动文摘系统 MEAD 应用于句子的检索与新信息过滤(Jahna et al.，2003)。在句子检索方面，他们在自动文摘系统中融入如下指标。

(1) Centroid：句子到句子集合中心向量的距离。

(2) LexRank：用来表示句子的重要程度，最早在多文档自动文摘中提出。为了计算句子的 LexRank，需要先要构造一个无向图(其中每个节点是一个句子)。如果任意两个句子的余弦相似度大于某个预设的阈值，则将对应的两个节点用一条边连接，边的权重需归一化处理，使得每个节点出边的权重之和为 1。采用矩阵表示，各个数据都是一个统计量，因此可将该图看做是离散的马尔可夫链，一个句子的 LexRank 值就是它在马尔可夫链中相应的静态分布概率。直观地说，如果一个句子和大量的其他句子相似，那么其 LexRank 值会比较高，同时和它相似的句子也会有一个较高的 LexRank 值。

(3) Length:句子的词数。

(4) QueryTitleCosine:句子与主题中标题部分的余弦相似度。

(5) QueryDescriptionCosine:句子与主题中简介部分的余弦相似度。

在自动文摘的结果基础上,可以进一步组织,构建出相关的句子结果。

2.11 句子级新信息检测

句子级新信息检测的核心问题在于如何在句子序列基础上度量某个句子所含信息的新颖度,这需要对两个不同句子之间的信息冗余度进行度量。信息冗余度量化之后,就可以根据预设的阈值,检测出新信息。研究者提出了多种技术手段对信息冗余度进行量化,下面是几种常用的信息新颖度计算方法。

2.11.1 词重叠度

将两个句子分别看成词集合 A 和 B(A 在 B 前)。最简单的信息冗余度可以量化为 $|A\cap B|$,Zhang 等(2002)定义了一种相对的词重叠度概念,即

$$\text{Overlap}B_A=\frac{A\cap B}{B}$$

因此,B 的词重叠度就可以定义为 $\max\{\text{Overlap } B_i \mid i \text{ 为 } B \text{ 之前的句子}\}$。

2.11.2 最大区间相关度

最大区间相关度最早由 Carbonell 和 Goldstein(1998)提出。在新信息检测方面,主要兼顾当前句子与查询和以前句子的相关度,并寻找两者区间最大的结果。文献(Sun et al.,2003)对 MMR 的定义为

$$\text{MMR}(s_j)=\lambda\text{Sim}_1(V_{s_i},V_{\text{tp}})-(1-\lambda)\max_{S\in R}\text{Sim}_2(V_{s_i},V_{s_j})$$

其中,s_i 为当前句子;V_{s_i}、V_{tp}、V_{s_j} 分别为当前句子、主题和以前句子的向量表示形式;R 为 s_i 之前的相关句子集合;Sim_1 为句子和主题相似度计算函数;Sim_2 为句子之间的相似度计算函数;λ 为两者之间的调节参数($0\leqslant\lambda\leqslant1$),一般设为 0.7。

2.11.3 余弦冗余度

两个句子 s_i 与 s_j 可以采用计算向量之间的余弦相似度,即

$$\cos(s_i,s_j)=\frac{\sum_{k=1}^{n}v_{i,k}\times v_{j,k}}{\|s_i\|\cdot\|s_j\|}$$

由于句子间的相似度实质上也是两个句子之间的信息冗余度(Tsai and Chen,2002),因此可以采用余弦相似度来计算当前句子与以前句子的

信息冗余度，在排除信息冗余度大于一定阈值的句子后，一般来说就可以获得新信息。

2.11.4　命名实体触发方法

也有学者采用一些启发式的规则，如新人物、新机构或者新地方往往意味着有新信息内容。因此，可得到一条启发式规则，即出现了新的命名实体的句子往往包含了新的信息。Abdul 等(2004)认为如果句子中出现的人名、机构名或者其他命名实体是以前没有见过的，则断定该句包含了新信息，并因此在新信息检测系统中引入了 20 多种命名实体，包括人名、地名、机构名、日期以及金钱等。命名实体识别是经典的信息抽取问题，也是 MUC(Message Understanding Conference)大会主要的评测任务之一。

2.11.5　统计机器翻译模型

卡内基梅隆大学(CMU)提出一种基于统计机器翻译模型的新信息检测方法(Thompson et al.，2002)，基本思想为将冗余句子视为第一个句子的统计机器翻译结果，在这种语言上如可构建一个好的翻译模型，那么系统就可检测出这两个句子是否翻译了同样的内容，进而发现新信息内容。在新信息检测的应用中，源语言(待翻译的语言)与目标语言(翻译结果的语言)都是同一种语言，因此问题就大大简化了。可借助 WordNet 来估计词语与短语的相似度，同时采用浅层句法分析来抽取并比较不同句子的结构。给定两个待比较的句子，算法处理过程主要包括下面几个阶段：首先获得每个句子的句法分析树；其次将每棵句法分析树转化为图的形式(要求该图可以表达出各个词语修饰成分的结构)；再次实现一个简单的图匹配算法，从而比较句子中的各个词语(其中词语权重主要表示可能的重要度)；最后计算出一个句子在同一语言上翻译为另一个句子的概率(可视为两个句子之间的相似度)。下面是两个从 TREC 训练文档中抽取的例句。

Sentence *A*：Some of the best shots，**released** this month by the US space agency Nasa，**show** parts of the universe billions of light years away-and therefore **billions** of **years** in the **past**.

Sentence *B*：The images sent back this **year**，after astronauts repaired the telescope's defective mirror，**show** a **myriad** of astronomical objects too distant to be seen with the most **powerful** Earth-bound observatories.

如采用统计机器翻译模型，按照上面的过程计算句子 A 和 B 的相似度：先抽取相似度最高的六个词对(表 2.1 给出了计算的中间结果)，其中第三列给出的是句子 B 中与对应词相似距离最小的词，如与句子 A 中的“past”相似度距离最小的词为“year”。相似距离计算主要采用 WordNet 并结合统计信息。最后这两个句

子的相似距离为这六个词对的平均权重(结果为 14.821,小于设定的经验阈值 15),因此可以判定句子 B 为冗余信息。

表 2.1 采用统计机器翻译模型计算的句子 *A* 与 *B* 的信息相似度

Sentence A	Graph weight A_i	Sentence B (Most similar word)	Graph weight B_i	Similarity Distance S_i
past	6	year	4	2.456
years	3	year	4	0.0258
released	4	show	2	53.631
Nasa	5	powerful	1	68.240
show	2	show	2	0.000
billions	3	myriad	1	0.152
Weighted mean:14.821				

2.11.6 LexRank 方法

在句子级检索的自动文摘方法中简单介绍了 LexRank。文献(Erkan,2004)采取 LexRank 来对当前句子与以前句子的冗余度进行建模。首先按照相关句子向量之间的余弦值建立相似句子有向图,边的权值为节点之间的相似度,图中边的方向表示的是句子之间的时序关系。图 2.10 给出了相似句子有向图示例,其中句子 1 与句子 2 和句子 4 相似,句子 2 与句子 3 相似。

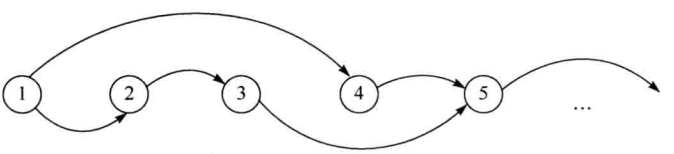


图 2.10 采用 LexRank 表示的相似句子有向图

其次,为了评估每个句子节点的新信息量,在有向图上计算 LexRank 值。在理想情况下,LexRank 值较小的句子应包含新信息(因 LexRank 值较小意味着句子基本上没有入边或者入边很少,也就是说,它和以前的句子基本上都不相似,反之亦然),因此在图 2.10 中句子 5 包含新信息的概率非常小,基本可以断定这是一个冗余句,而第一个句子没有入边,它的 LexRank 值总是很小,因此它肯定是新信息句子。最后,依靠 LexRank 值,采取简单的决策算法就可预测出所有的新信息结果。

2.12　本章小结

本章对大数据搜索与挖掘领域中的自然语言处理中的关键技术，如词法分析中的自动分词、未登录词识别、有意义串识别等方法，进行了综述，并对词典组织、基于上述技术的文本分类与聚类、话题识别与跟踪技术进行了介绍。

相关工作与扩展阅读

张华平(2005)提出了一种将汉语分词、词性标注、切分排歧和未登录词识别相结合的基于层次隐马尔可夫模型的理论框架。在语言分析的基础上，查询分析过程通过辅助词过滤与倾向分类，从自然语言表述的主题中理解用户的查询意图，从而抽取出可用于直接建模计算的查询向量。在句子检索方面，原型系统 Noovel 采用向量空间模型、概率检索模型与语言模型。为了克服句子的局限性，引入了查询扩展技术，借助于 WordNet 的语义衍生扩展、伪相关反馈扩展、采用高频共现词语的局部共现扩展。Noovel 系统参加了第 13 届 TREC 比赛新信息发现任务的全部四个子任务，在最关键的任务 1 中，Noovel 的新信息检测结果排名第一；任务 3 的句子检索性能方面，提交的两个结果并列排名第一，其他的子任务也取得了不俗的成绩。于满泉(2006)通过数据预处理，把文本内容转化成人物模型的信息片段；研究了人物模型的同一性判断方法和数据融合方法，通过处理人物模型片段的信息，不仅要把不同人物的信息区分开，还要把同一人物的信息进行整合，形成完整的人物生平履历。在人物模型同一性判断的基础上，针对同一人物的活动报道，研究了人物活动事件的组织方法，探讨了人物追踪若干细粒度挖掘任务的处理方法，并以流通度理论和动态流通语料库理论为基础，研究了人物知名度的计算方法，通过跟踪人物不同时期的知名度变化，绘制出人物知名度变化的曲线图。贺敏(2007)提出了一种上下文邻接分析和语言模型分析相结合的有意义串提取算法，通过重复串发现得到候选串，计算邻接特征量来衡量候选串是否满足语用多样性，采用熟语料上训练好的双字耦合度和单字位置成词概率来判断候选串的语义完整性，并对实验产生的有意义串进行了语言学分析。黄玉兰(2009)发现有意义串遵循空间局部性原理和时间局部性原理，并用 LE 值量化字符串的局部性，在此基础上利用 LE 值和其他统计量共同提取有意义串。另外，该文提出了基于伪相关反馈模型的有意义串提取算法，将有意义串的自动生成过程视为一个信息检索过程，采用检索模型中常用的伪相关反馈技术，进一步提高了有意义串提取的准确率。刘治华(2011)在提高摘要的处理效率、基于关键词提取摘要、摘要中句子间的冗余去除方面进行了研究。在提高效率方面，引入搜索引擎的倒排链表结构统计词和句子的特征，并使用双数组 Trie 树存储分词词典和用户主题词表，以期提高词的查找效

率，将关键词提取与摘要提取相结合，在关键词提取中引入了词邻接类别、词的位置局部性分别提高高频词、低频词的质量。在去除句子冗余度方面，提出句子之间的包含度概念，能够对文章中存在包含关系的一类句子进行有效的排重，降低文摘的冗余度，提高文摘质量。王思力等(2006)结合实验结果综合分析了分词系统对中文信息检索的影响，在此基础上系统地提出了适用于信息检索应用的分词技术特点，考虑到用于信息检索的分词算法对速度要求比较高，所以词典算法采用双数组 Trie 树优化算法。该算法查询一个词语最多只需要进行 $n-1$ 次整数相加，时间复杂度是$O(n)$，n为查询词长度。对于影响切分精度的两大技术难点(即歧义消除和未登录词识别)，分为两步分别处理：对歧义消除，在分词阶段不处理覆盖歧义，保留到查询扩展时处理；对于交叉歧义，用双字耦合度和 t-测试差结合的方法来消歧。对于未登录词识别，采用单字的成词位置概率，同时结合局部二元串频统计对所有的命名实体和新词进行统一识别。李继峰(2003)提出了双层翻译模板的体系结构，并给出了一系列模板的定义，给出了从句子对齐的汉英双语语料库自动抽取双层翻译模板的算法，设计了利用双层模板库进行汉英翻译的高效率算法，根据这些结构和算法，实现了一个机器翻译原型系统。于江德(2012)指出现阶段汉语词法分析的主流技术是基于统计的方法，这类方法的本质是把词法分析过程看作序列数据标注问题。上下文是统计方法中获取语言知识和解决自然语言处理中多种实际应用问题必须依靠的资源和基础。汉语词法分析时需要从上下文获取相关的语言知识，但上文和下文是否同样重要呢？为克服仅凭主观经验给出猜测结果的不足，对基于字标注汉语词法分析的分词、词性标注、命名实体识别这三项子任务进行了深入研究，对比了上文和下文对各个任务性能的影响，结果表明在字标注框架下，下文对汉语词法分析性能的贡献比上文的贡献高出 6 个百分点以上。

参 考 文 献

卜东波，白硕，李国杰．2002. 文本聚类中权重计算的对偶性策略．软件学报，13(11)：2083-2089.

高凯，郭立炜，许云峰．2010. 网络信息检索技术及搜索引擎系统开发．北京：科学出版社.

贺敏．2007. 面向互联网的中文有意义串挖掘．中国科学院计算技术研究所博士学位论文.

呼声波，刘希玉．2008. 网页分类中特征提取方法的比较与改进．山东师范大学学报(自然科学版)，23(3)：35-37.

胡吉祥．2006. 基于频繁模式的消息文本聚类研究．中国科学院研究生院学位论文.

黄玉兰．2009. 有意义串挖掘及其应用．中国科学院计算技术研究所硕士学位论文.

贾自艳，史忠植．2004. 基于概率统计技术和规则方法的新词发现．计算机工程，30(20)：19-21，83.

姜维，王晓龙，关毅，等．2007. 基于多知识源的中文词法分析系统．计算机学报，30(1)：

137-145.
焦玉英 . 2003. 信息检索进展 . 北京：科学出版社.
李晗静，李生，赵铁军，等 . 2007. 基于自然语言理解的实体自动摆放的研究 . 电子与信息学报，29(8)：1845-1849.
李继峰 . 2003. 基于双层模板结构的机器翻译方法研究 . 中国科学院计算技术研究所硕士学位论文.
李庆虎，陈玉健，孙家广 . 2002. 一种中文分词词典新机制——双字哈希机制 . 中文信息学报，17(4)：13-18.
李振星，徐泽平，唐卫清，等 . 2003. 网页多词元快速聚类算法 . 计算机工程，29(2)：20-22.
凌云，王勋，费玉莲 . 2003. 智能技术与信息处理 . 北京：科学出版社.
刘治华 . 2011. 面向主题的文档摘要技术研究 . 北方工业大学硕士学位论文.
苗夺谦 . 2007. 中文文本信息处理的原理与应用 . 北京：清华大学出版社.
沈达阳，孙茂松，黄昌宁 . 1995. 中国地名的自动辨识 . 北京：清华大学出版社.
宋柔，朱宏，潘维桂，等 . 1993. 基于语料库和规则库的人名识别法 . 计算语言学研究与应用：150-154.
孙建军，成颖 . 2004. 信息检索技术 . 北京：科学出版社.
孙茂松，张维杰 . 1993. 英语姓名译名的自动辨识 . 计算语言学研究与应用 . 北京：北京语言学院出版社.
孙茂松，黄昌宁，高海燕，等 . 1995. 中文姓名的自动辨识 . 中文信息学报，9(2)：16-27.
孙茂松，左正平，黄昌宁 . 2000. 汉语自动分词词典机制的实验研究 . 中文信息学报，14(1)：1-6.
王开铸，李俊杰，吴岩 . 1995. 无词典自动分词的研究 . 计算语言学进展与应用，13(2)：35-38.
王思力 . 2006. 面向大规模信息检索的中文分词技术研究 . 中国科学院计算技术研究所硕士学位论文.
王思力，张华平，王斌 . 2006. 双数组 Trie 树算法的优化及其应用研究 . 中文信息学报，20(5)：24-30.
王晓龙，关毅 . 2005. 计算机自然语言处理 . 北京：清华大学出版社.
王永成 . 1991. 中文信息处理技术及其基础 . 上海：上海交通大学出版社.
徐宝文，张卫丰 . 2003. 搜索引擎与信息获取技术 . 北京：清华大学出版社.
尹中航 . 2002. 网络新闻智能分类技术的研究与实现 . 上海交通大学硕士学位论文.
于江德 . 2012. 字标注汉语词法分析中上文和下文孰重孰轻 . 计算机科学，39(11)：201-203，236.
于满泉 . 2006. 面向人物追踪的知识挖掘研究 . 中国科学院计算技术研究所硕士学位论文.
俞士汶 . 2004. 计算语言学概论 . 北京：商务印书馆.
张华平，李恒训，刘治华 . 2010. 信息检索：算法与启发式方法 . 北京：人民邮电出版社.
张华平，刘群 . 2004. 基于角色标注的人名识别 . 计算机学报，27(1)：85-91.
张华平 . 2005. 语言浅层分析与句子级新信息检测研究 . 中国科学院计算技术研究所博士学位论文.

张勇，何婷婷 . 2005. 基于质子串分解的网络新词汇自动抽取//全国第八届计算语言学联合学术会议(JSCI-2005)论文集.

赵军，金千里，徐波 . 2005. 面向文本检索的语义计算 . 计算机学报，28(12)：2068-2078.

郑家恒，李文花 . 2002. 基于构词法的网络新词自动识别初探 . 山西大学学报(自然科学版)，25(2)：115-119.

邹刚，刘群 . 2004. 面向 Internet 的中文新词语检测 . 中文信息学报，18 (6)：1-9.

Abdul J N, Allan J, Croft W B, et al. 2004. UMass at TREC 2004: novelty and hard// Proceedings of the 13th Text Retrieval Conference.

Baeza-Yates R, Ribeiro-Neto B. 1999. Modern Information Retrieval. New York: ACM Press.

Bao S H, Zhang L, Chen E D, et al. 2006. LSM: language sense model for information retrieval// Proceedings of the 7th International Conference on Web-Age Information Management.

Bikel D M, Miller S, Schwartz R, et al. 1997. A high-performance learning name finder// Proceedings of the 5th Conference on Applied Natural Language Processing.

Brants T, Chen F R, Farahat A O. 2003. A system for new event detection// Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.

Carbonell J, Goldstein J. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries// Proceedings of SIGIR.

Carthy J. 2002. Lexical chains for topic tracking. PhD Thesis of National University of Dublin.

Charniak E. 1993. Statistical Language Learning. New York: MIT Press.

Chen A T. 2003. Chinese word segmentation using minimal linguistic knowledge// Proceedings of the Second SIGHAN Workshop on Chinese Language Processing.

Chen K J, Bai M H. 1988. Unknown word detection for Chinese by a corpus-based learning method. International Journal of Computational Lingustics and Chinese Language Processing, 3(1): 27-44.

Chen Y, Chan K P. 2007. Using data mining techniques and rough set theory for language modeling//ACM Transactions on Asian Language Information Processing.

Conroy J M. 2004. A hidden markov model for the TREC novelty task// Proceedings of the 13th Text Retrieval Conference.

Conroy J M, Daniel M, Dianne P. 2003. From TREC to DUC to TREC// Proceeding of the 20th Text Retrieval Conference.

David A S. 2003. Detecting and browsing events in unstructured text// Proceedings of the 25th Annual ACM SIGIR Conference: 73-80.

Erkan G. 2004. The university of michigan in novelty 2004// Proceedings of the 13th Text Retrieval Conference.

Feolman R. 2007. The Test Mining Handbook: an Approaches in Analyzing Unstructured Data. Cambridge: Cambridge University Press.

Forman G. 2006. Tackling concept drift by temporal inductive transfer// Proceedings of the ACM

SIGIR Conference.

Franz M, McCarley J S, Roukos S, et al. 1999. Segmentation and detection at IBM: hybrid statistical models and two-tiered clustering// Proceedings of the TDT Workshop.

Gao K, Wang Y C, Wang Z Q. 2004. An efficient relevant evaluation model in information retrieval and its application// Proceedings of the 4th International Conference on Computer and Information Technology.

Han J W, Micheline K. 2001. Data mining concepts and techniques. 北京：机械工业出版社.

Hand D. 2006. Classifier technology and the illusion of progress. Statistical Science, 21: 1-14.

Hideto K, Tsutomu H, Hideki I, et al. 2002. A machine learning approach for QA and novelty tracks: NTT system description// Proceedings of the 11th Text Retrieval Conference.

Hodjat B, Franco H, Bratt H, et al. 2003. Iterative statistical language model generation for use with an agent-oriented natural language interlace// Proceedings of the 10th International Conference on Human-Computer Interaction.

Hotho A, Stumme G. 2002. Conceptual clustering of text clusters// Proceedings of FGML Workshop Special Interest Group of German Informatics Society.

Jahna O, Hong Q, Ali H, et al. 2003. The university of michigan at TREC 2003// Proceeding of the Eleventh Text Retrieval Conference.

Jain A K, Dubes R C. 1988. Algorithms for Clustering Data. New York: Prentice Hall Press.

Joachims T. 2006. Training linear SVMs in linear time// Proceedings of the International Conference on KDD.

Lai Y S, Wu C H. 2002. Meaningful term extraction and discriminative term selection in text categorization via unknown-word methodology//ACM Transactions on Asian Language Information Processing, 1(1): 34-64.

Lee F C. 1997. Pat-tree-based keyword extraction for Chinese information retrieval// Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.

Lee J H, Kim W Y, Kim M H, et al. 1993. On the evaluation of boolean operators in the extended boolean retrieval framework// Proceedings of the 16th ACM International Conference on SIGIR.

Lewis D. 2004. A comparison of two learning algorithms for text categorization// Proceedings of the SDAIR Conference: 81-93.

Li H Q. 2004. The use of SVM for Chinese new word international// Proceedings of the Joint Conference on Natural Language Processing.

Losee R M, Bookstein A. 1988. Integrating boolean queries in conjunctive normal form with probabilistic retrieval models. Information Processing and Management, 24(3): 315-321.

Makkonen J, Helena A M, Marko S. 2004. Simple semantics in topic detection and tracking. Information Retrieval, 7 (3,4): 347-368.

Meng Y, Hao Y. 2004. Chinese new word identification based on character parsing model// Pro-

ceedings of the First International Joint Conference on Natural Language Processing.

Mikheev A, Grover C, Moens M. 1998. Description of the LTG system used for MUC-7// Proceedings of the Seventh Message Understanding Conference.

Minwoo J, Lee G G. 2008. Improving speech recognition and understanding using error-corrective reranking//ACM Transactions on Asian Language Information Processing.

Pantel P, Dekang L. 2002. Document clustering with committees// Proceedings of SIGIR.

Peng F C, Fang F. 2004. Chinese segmentation and new word detection using conditional random fields//Proceedings of the 20th International Conference on Computational Linguistics.

Ponte J M, Croft W B. 1998. A language modelling approach to information retrieval// Proceedings of the 21st ACM SIGIR Conference on Research and Development in Information Retrieval: 275-281.

Rosenfeld R. 2000. Two decades of statistical language modeling: where do we go from here// Proceedings of the IEEE.

Salton G A. 1969. Comparison between manual and automatic indexing methods. Journal of American Documentation, 20 (1) : 61-71.

Salton G. 1970. Automatic text analysis. Science, 168(3929): 335-342.

Salton G. 1971. The Smart Retrieval System: Experiments in Automatic Document Processing. New York: Prentice-Hall Press.

Salton G, Edward A F, Wu H. 1983. Extended boolean information retrieval. Communication of the ACM, 26(11): 1022-1036.

Sekine S, Grishman R, Shinou H. 1998. A decision tree method for finding and classifying names in Japanese texts// Proceedings of the Sixth Workshop on Very Large Corpora.

Sun J, Pan W F, Zhang H P. 2003. TREC 2003 novelty and web track at ICT// Proceeding of the Twelfth Text Retrieval Conference.

Taoufiq D, Mothe J. 2004. TREC novelty track at IRIT-SIG// Proceedings of the 13th Text Retrieval Conference.

Thompson K C, Paul O, Zhang Y, et al. 2002. Information filtering, novelty detection, and named-page finding// Proceedings of the 11th Text Retrieval Conference.

Tsai M F, Chen H H. 2002. Some similarity computation methods in novelty detection// Proceeding of the 11th Text Retrieval Conference.

Wayne C. 2000. Multilingual topic detection and tracking: successful research enabled by corpora and evaluation// Proceedings of the Language Resources and Evaluation Conference (LREC): 1487-1494.

Wong S K M, Ziarko W, Wong P C N. 1985. Generalized vector space model in information retrieval// Proceedings of 8th ACM International Conference on SIGIR.

Wu A D, Jiang Z X. 2000. Statistically-enhanced new word identification in a rule-based Chinese system// Proceedings of the Second Chinese Language Processing Workshop: 46-51.

Yang Y M. 1999a. Are-examination of text categorization methods// Proceedings of ACM SIGIR

Conference on Research and Development in Information Retrieval.

YangY M. 1999b. An evaluation of statistical approaches to text categorization. Journal of Information Retrieval, 1(2): 67-88.

Yang Y M, Jaime C, Ralf B, et al. 1999. Learning approaches for detecting and tracking news events. IEEE Intelligent Systems: Special Issue on Applications of Intelligent Information Retrieval, 14(4): 32-43.

Yang Y, Carbonell J, Jin C. 2002. Topic-conditioned novelty detection// Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Zhang J, Gao J F, Zhou M. 2000. Extraction of Chinese compound words-an experimental study on a very large corpus// Proceedings of the ACL Chinese Language Processing Workshop.

Zhang M, Song R, Lin C, et al. 2002. Expansion-based technologies in finding relevant and new information: TREC2002 novelty track experiments// Proceeding of the Eleventh Text Retrieval Conference.

第 3 章　大数据检索与分词

3.1　概　　述

在中文自然语言中，词是最小的能够独立活动的有意义的语言成分。但是汉语词语之间没有明显的区分标记，因此在进行中文自然语言处理前，通常是先将汉语文本中的字符串切分成合理的词语序列，再在此基础上进行其他分析处理。将中文字符串切分成合理词语序列的过程就是中文分词，它是中文信息处理的一个基础环节，已经被广泛应用于中文文本处理、信息提取和文本挖掘等应用中。相关学者和研究人员在这个领域已经取得了一些成果，有一些实用的通用分词算法。在 Lucene 中也集成了一些通用的分词算法，如在 Lucene 中，Analysis 包完成分词。Analysis 包的主要功能包括将文本词汇单元化（包括去除标点、字母规格化、移除常用词、Stemming 等）、将各词汇单元与关联的域名相结合形成各个词项等。虽然词法分析实现起来比较复杂，但 Lucene 巧妙地避开了这个领域，转而使用 JavaCC 为其构造标准的分词器。JavaCC 是一种内置标准语言解析器的生成工具，表示 Java Compiler Compiler，它是对著名的词法分析工具 YACC（Yet Another Compiler Compiler）和词法分析器 Lex 的继承，只不过 YACC 生成的是 C 代码，而 JavaCC 生成的则是 Java 代码。例如，在 Lucene 的 org. apache. Lucene. analysis. standard中的类，就是 JavaCC 为 Lucene 准备的，其中的 StandardTokenizer. java 继承了 Lucene 的 Tokenizer，并担当起对传入的 Reader 进行词条切分的任务，StandardTokenizer. next()方法可以按照在 JavaCC 的 .jj 文件中定义的词条样式切分出下一个词条。为此，Lucene 专门设计了一个标准的分析器 StandardAnalyzer 来使用这个 StandardTokenizer。对上述各种分词结果的分析，可参见相关文献（高凯等，2010）。

分词算法要想获得更好的切分精度，通常需要利用更多的语言资源，这样必然耗费更多的时间去处理这些语言资源，因此对于一般切分精度比较高的算法，其切分的速度是比较慢的。一些切分速度快的算法，因为抛弃了一些繁琐的语言处理，故切分精度一般不高。从当前的一些主要分词算法来看，切分精度虽然有差别，但差异都不是特别大，而切分时间相对来说差别则比较大。根据我们的实验，在相同的实验环境下，较快的切分算法（如最大匹配算法）可以达到 20MB/s 以上，而慢的切分算法（如基于层次隐马尔可夫模型的分词算法）只有 100KB/s 左右。对于当前的一些实际应用，如 Web 搜索，时间效率是重要的。那些较高切分精度的分词

算法，因为速度太慢不仅无法满足实际应用的需求，有时甚至也无法满足其他自然语言处理研究的需要。在目前的许多实际应用产品中，为了保证速度都不得不牺牲部分准确度，可能也采取了一些比较简单的切分算法。

限于当前的技术水准，汉语切分要做到百分之百的正确是不太可能的，而且由于语言本身的特点，词语没有确定的界定标准。怎么样才算是词或词组，一直没有统一确定的标准，这也导致了上层不同的信息处理对分词标准有不同的要求。对于大数据检索系统来说，倾向于分词单位较小化。例如，把“并行计算机”切成“并行/计算机”，把“计算语言学”切成“计算/语言学”，使得无论用“并行计算机”还是用“计算机”、“计算语言学”或是“语言学”检索，都能查到。这时候分词单位元的粒度大小就需要考虑到查全率和查准率的矛盾。

近年来，伴随着 SIGHAN 国际中文分词评测活动 Bakeoff 的开展，中文分词技术有了长足的进步，在分词方法和理念上都有了很大的创新。有的学者运用标注的思想把中文分词问题转换为序列标注问题，然后使用最大熵的模型来进行字的标注，2003 年 SIGHAN Bakeoff 中文分词比赛中获得了较高的分词精度和很高的未登录词召回率。自此，基于字的标注引起了广泛的关注。在此后两届的 SIGHAN Bakeoff 中，在分词上达到很好效果的大都采用了基于字的标注思想。基于子词的分词模型不仅能够有效地利用字本身的上下文特征，而且能够有效融合字与词的组合特征。基于子词的模型是在基于字的标注模型下建立起来的，仍然采用序列标注的思想，不同的是将基于字的模型泛化为基于子词的模型，把子词作为最小的切分单元来看待。

学术界关于分词的评测多以通用的分词精度为准绳。鉴于分词在切分标准和切分算法本身上存在的诸多困难，我们希望能够针对一项具体的上层应用来研究相关的分词技术，这样便于有一个比较确定的分词规范和目标，然后可以有针对性地在分词算法方面有所突破。由于 Web 搜索是跟人们生活最接近、应用最频繁而且技术发展也最成熟的一项信息处理技术，因此我们选择了基于大规模中文信息检索中的分词算法并给出部分研究成果。

本章首先介绍中文分词技术的难点以及目前常用的一些切分算法，然后结合实验结果，分析分词对中文信息检索的影响，在此基础上提出适用于信息检索的分词技术的特点，并根据这些特点给出相应的算法。它是基于词典匹配和统计方法，所用的词典采用双数组 Trie 树优化算法。

3.2　分词对中文信息检索的影响

在中文信息检索中，如果不分词，用作标引的就是单个汉字或是按 N 元切分法进行切分后得到的切分单位。那么，是否需要按语言学意义上对词进行切分？

文档和查询二者的切分方法是否需要一致？对基于字的检索和基于词的检索，哪个性能更好？不同的研究者都有不同的实验结果。Schubert 和 Li(2004)给出参加 TREC5 和 TREC6 中文信息检索实验的一些研究小组的实验结果(王思力,2006)。从统计分析中可以看出,7 组中有 3 组的结论是基于字比基于词的切分方法好,有 3 组结论相反,还有 1 组认为二者差不多;7 组中有 2 组认为二者结合起来效果要比单独用一种方法更好。在 5 组比较切分方法一致性的实验中,有 3 组显示切分方法一致的实验结果更好,另外 2 组切分方法不一致的结果则是矛盾的:1 组文档用基于字的切分和最大匹配切分,查询用基于字的切分,结果显示最大匹配的结果更好。另一组文档用单字切分和基于词的最大匹配切分,查询用基于词的切分,结果显示基于单字的效果更好。可以看到,单凭上面的实验结果很难下结论说到底哪种切分方法更好,也很难说文档切分和查询切分方法是否应该保持一致。

我们先分析一下出现这些结果的原因,然后再进行实验证明我们的分析结果。在上述 7 组实验结果中,结论为“字切分”比“词切分”好的实验一共有 3 组,其中的“词切分”方法有两组都是最大匹配。我们知道,基于词典的最大匹配有三个明显的缺点:一是切分准确性依赖于词典的规模,对于词典中没有的词无法正确切分;二是不能很好地处理歧义问题;三是匹配出来的都是最长词,标引得到的倒排索引文件中都以最长词为索引。这无法确保文档和查询的灵活匹配,因为用户提交的查询和文档经常会用不完全相同的一个词来表示同一类概念,如果用户提交的是一个短的查询词,则无法查到出现此短词和其他词组合成词的文档。例如,对于包含词“中国农业科学院”的文档,用最大匹配法去切分时,会被切分成“中国\农业科学院”,如果用户想查找“中国的科学院”相关的文档,那么所有包含“中国农业科学院”的文档都无法被检索到。因此,这里假定之所以“基于词”的检索性能比“基于字”的低,是因为最大匹配法切分准确率过低且切分粒度过大。如果因此就说“基于字”的切分方法比“基于词”的方法更适合信息检索,恐怕言之过早;文档切分和查询切分一致与不一致的实验结果,则跟所使用的具体分词算法有关,所以同样不能下结论。为此,我们设计如下两组实验。

第一组实验方案:基于单字切分、交叉二元切分和利用 ICTCLAS 系统切分的检索性能比较。文档和查询采用同一种切分方法。

第二组实验方案:基于单字切分、交叉二元切分和利用 ICTCLAS 系统切分的检索性能比较。查询采用人工切分的方法。

在这两组测试所用的三种切分方法中,前二种是基于字的切分,第三种是基于层次隐马尔可夫模型的分词方法。为了减少最长词切分对检索的影响,我们在第三种方法中去掉了词典中的一些长词。从分词准确度上来说,ICTCLAS 曾经在第一届 SIGHAN 评测中获过两个单项第一,分词精度要比前两种高很多。这两组

实验方案的第一组在于比较基于字切分和基于词切分对检索性能影响的大小，第二组在于比较文档和查询切分一致和不一致之间的优劣。我们选取了部分查询主题，采用人工切分的方法和第一组的实验结果做比较。测试数据是北京大学提供的中文网页测试集 CWT，该测试集共 90G 数据，分为主题提取和导航搜索两个任务。我们只选择了主题提取任务，该部分提供 60 个主题。受实验软硬件环境的限制，只抽取部分测试集数据进行测试。检索系统利用马萨诸塞大学和卡内基梅隆大学合作开发的检索工具包 Lemur。

针对实验方案一的结果如表 3.1 所示，结果表明当分词算法精度比较高的时候，基于词的分词算法要比基于字的切分方法对检索性能的贡献高。

表 3.1　三种切分方法在文档和查询切分方法一致情况下的实验结果

	单字	二元交叉切分	ICTCLAS
召回率	插值召回-准确率平均值		
0.0	0.7429	0.8100	0.7534
0.1	0.5745	0.6137	0.5923
0.2	0.4758	0.5222	0.5359
0.3	0.4068	0.4842	0.5150
0.4	0.3513	0.3930	0.4623
0.5	0.3015	0.3446	0.3893
0.6	0.2709	0.3047	0.2952
0.7	0.2326	0.2362	0.2435
0.8	0.1874	0.1777	0.1960
0.9	0.1293	0.1148	0.1070
1.0	0.0280	0.0252	0.0077
索引相关文档的评价准确率(非插值)			
	0.3192	0.3427	0.3583
准确率			
At 10 docs	0.4316	0.4895	0.5105
At 20 docs	0.4158	0.4421	0.4368
At 30 docs	0.3895	0.4211	0.4175
确率			
	0.3565	0.3823	0.3872

针对实验方案二的结果如表 3.2 所示。

表 3.2 文档切分和查询切分之间关系比较的实验结果

查询切分方法 / 文档切分方法	人工切分 (Average precision/ R-Precision)	和文档切分方法一致 (Average precision/ R-Precision)
单字切分	0.3021/0.3511	0.3192/0.3565
交叉二元切分	0.3312/0.3756	0.3427/0.3823
ICTCLAS	0.3496/0.3812	0.3583/3872

从表 3.2 有关实验方案二的结果可以看出，三种切分方法都是在查询和文档切分一致的情况下检索结果要好些。分析实验结果也发现，虽然人工切分保证了准确率，但因为分词算法的局限，自动分词必然存在错误，所以有的时候虽然查询切分正确了，但因为文档切分错误，还是不能检索到。如果不用人工切分而改用其他方法切分查询，同样也会存在同一个词被切成不同结果的情况。

在检索系统中，结果合并要占用很多时间。如果切分的粒度过小，在结果合并时就要花费更多时间。对于基于字切分的检索系统来说，在查询表达式比较长且二字词较少的情况下，检索效率要比基于词切分的检索系统低。例如，查询“清华大学”，如果是基于词的切分，在倒排索引中只需要查找“清华大学”这一个词的所有记录；如果是基于交叉二元切分，该查询被切分成“清华\华大\大学”三个词，在倒排索引中需要查这三个词的所有记录，然后再进行合并。这样基于字切分的查询和结果处理时间相当于基于词切分的 3 倍，当数据规模比较大的时候，这个时间差距会更加明显。

所以，综合检索精度和时间性能，具备较高精度的词切分方法要比字切分方法更适用于检索应用。

3.3 分词精度与检索性能的关系

有些分词精度较低或切分粒度较大的算法，在有的实验中可能得到的检索性能要比基于字的切分方法低，高精度的分词算法性能要比基于字的切分方法对检索性能的贡献大。那么是否可以说，分词精度越高的检索系统性能就越好呢？

Peng 等(2002)对上述二者的关系做了讨论，采用三种分词算法，即基于词典的前向最大匹配法、基于文本压缩的 PPM 算法和基于 EM 的自监督算法。其

中，最大匹配算法用了两个不同的词典，分别得到 71%和 85%的切分准确率；对于 PPM 算法，通过控制用于训练的 n 元模型 n 的大小(2 或者 3)，分别得到 90%和 95%的准确率。对于自监督的学习方法，通过控制 EM 算法迭代次数和调整词典修剪策略，得到的准确率为 44%、49%、53%、56%、59%、70%、75%、77%。这样相当于一共能得到 12 组不同精度的分词结果。

检索采用 OKAPI 系统①，通过设置参数 K_d 为不同的值得到不同的项权重来控制检索的性能。测试数据是 TREC5 和 TREC6 的中文测试集，评测采用 TREC 标准，计算平均准确率和确率值，由 TREC 的评测程序来完成，实验结果如表 3.3 和图 3.1 所示。

表 3.3　$K_d=10$ 时平均准确率和确率值

部分精确率/%	TREC5	TREC6
44	0.2231/0.2843	0.3424/0.3930
49	0.2647/0.3259	0.3848/0.4201
53	0.2999/0.3376	0.4492/0.4801
56	0.3056/0.3462	0.4473/0.4727
59	0.3097/0.3533	0.4740/0.4960
70	0.3721/0.3988	0.5044/0.5072
71	0.3656/0.4088	0.5133/0.5116
75	0.3652/0.4000	0.4987/0.5097
77	0.3661/0.4027	0.4968/0.4973
85	0.3488/0.3898	0.5049/0.5047
90	0.3213/0.3663	0.4983/0.5008
95	0.3189/0.3669	0.4867/0.4933

① Okapi 系统是由伦敦城市大学开发的基于概率检索模型设计的检索系统。它使用在概率模型中成熟的 bm25 公式及其变形(bm2500、bm250)等作为求词项 term 权值的公式。Okapi 同样提供建立索引和查询的工具。

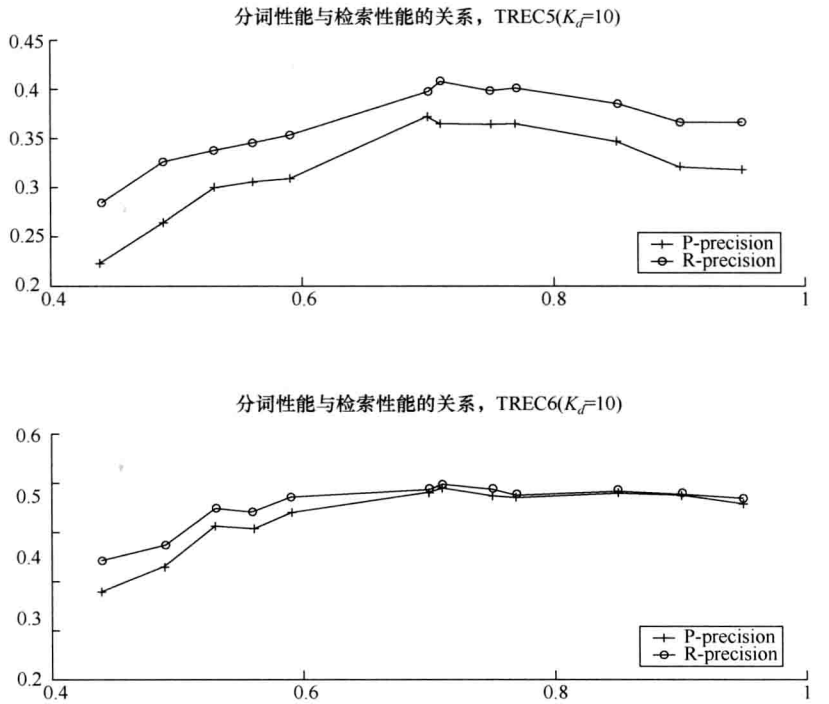
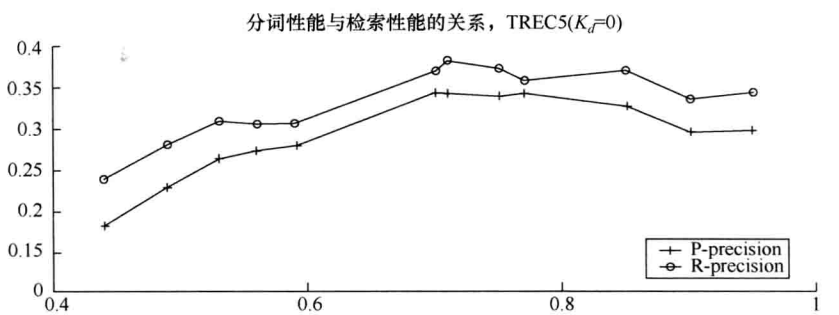


图 3.1 $K_d=10$ 时的 12 组检索结果比较

从图 3.1 中可以看出，$K_d=10$ 检索性能最好。检索性能并不总是随着分词精度的提高而提高。分词准确率到达某一个点之后，检索性能反而开始下降。当 K_d 等于其他值的时候，结果也同样如此，如图 3.2 和图 3.3 所示。



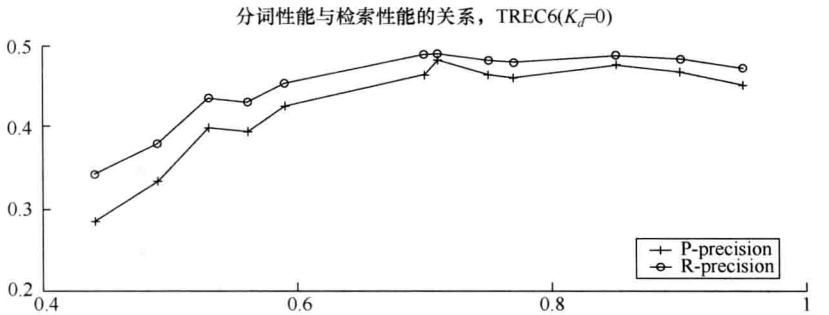


图 3.2　$K_d=0$ 时的 12 组检索结果比较

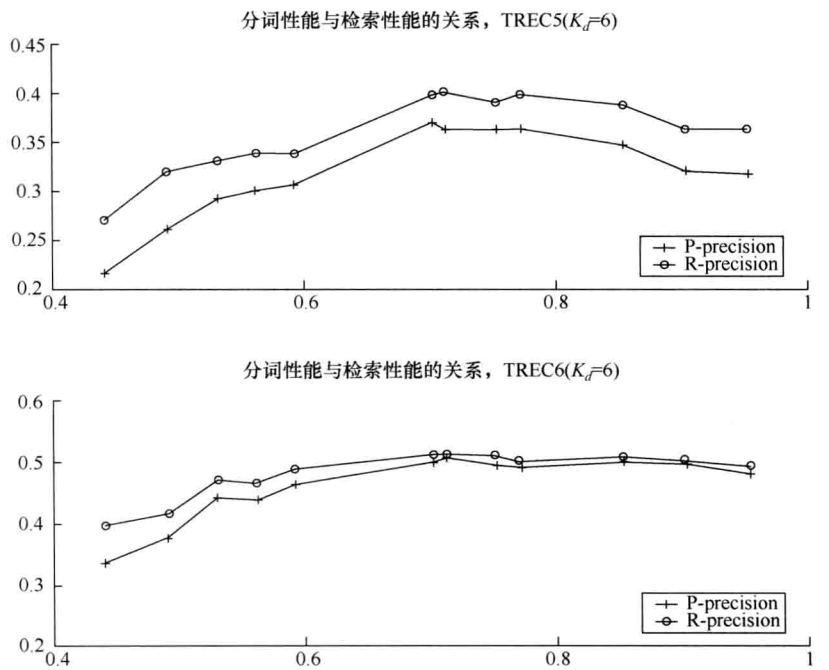


图 3.3　$K_d=6$ 时的 12 组检索结果比较

Schubert 和 Li(2004)认为在准确率较低时，可能分词标记对应的并不是一个语言学上的分词单位，也就是切分出来的不是一个真正的词语，所以部分检索性能会随着切分精度的增加而提高。当到达某个点后，检索性能不再随着分词精度的提高而提高，反而可能下降。该文认为一个可能的解释是切分精度较差的分词器会把一些混合词切分成较小的词单位，反而使得相关文档容易被检索到。在上述基础上，根据我们在检索实验中观察到的现象，归纳了检索性能不能随分词精度同

步递增的主要原因有如下几点。

(1) 因为查询切分和文档切分采用相同的分词算法，即使有一些切分错误的词，在查询时也遇到相同的切分，使得仍可正确检索到；相反，对于同一个词，有些高精度分词算法根据上下文信息在文档部分进行了正确切分，但是在查询部分可能因为缺乏相关的上下文信息而错误切分。可见，文档切分和查询切分结果的不一致，反而会导致检索准确率下降。

(2) 有些词被错误的切分成几个部分后，尽管这样会导致分词正确率下降，但对于检索来说，最后可以通过结果合并得到正确的结果，分词的错误并不影响检索的性能。

(3) 分词测得的准确率高低并不是绝对的，这涉及对词的定义问题。有些标准认为是该切分的词，实际上不切分时检索可能会更准确些。例如，“国内”在有的语料库中是两个词，但实际上切分成“国内”更准确一些。单独切开容易检索到一些不相关的文档；相反，对于那些按照有些标准认为是不该切分的词，实际上切分成较小的单位并用于检索，可能更准确些。

3.4 大数据应用环境下中文信息检索的分词算法及其特点

3.4.1 分词算法的时间性能要求高

有些中文分词算法的研究是以提高准确率为目标，可能采取了繁琐复杂的语言处理或者统计计算，导致一些高准确率的切分算法因为速度过慢而影响上层应用。这也是为什么有些分词算法上的新研究成果一直没能用于实际的原因之一。大规模的信息检索对速度非常敏感，尤其是现在的大数据搜索。作为中文信息处理基础的分词，必须占用尽可能少的时间，否则无论多高精确度的分词算法，也不方便用于实际的大规模信息检索系统。

3.4.2 分词正确率的提高并不一定带来检索性能的提高

虽然从理论上来说，准确率高的分词算法要比准确率低的分词算法带来更好的检索结果，但是在实际应用当中，由于受目前检索技术和自然语言处理技术的限制，准确率较低的分词算法并不一定检索结果就比准确率较高的分词算法的检索结果差。实验表明，分词算法的准确率与信息检索的查准率和召回率并不呈绝对的正比关系。实验证明，分词到达一定精度之后，对中文信息检索的影响不再明显，一味追求高准确率的分词算法，并不适合大规模中文信息检索。在时间和精度之间存在矛盾无法兼顾的情况下，需要在二者之间找到一个平衡。

3.4.3 分词切分粒度需在查询扩展层面进行相关处理

作为影响分词准确率的两大因素之一，歧义分为交叉歧义和覆盖歧义两种。

在信息检索中，分词算法应集中精力考虑如何消除交叉歧义；对于覆盖歧义，考虑到信息检索的特点，我们提出另外一种在检索层面上的解决方法，而不是在分词层面来解决它。

对覆盖歧义来说，可以利用词典的二次索引和查询扩展。对于词典中的每个词建立一个覆盖词的二次索引，如“中国科学院”，其覆盖词包括“中国”、“科学”和“科学院”。因此，对于“科学院”这样的查询请求，在查询扩展时就可以加入“中国科学院”，最后返回的结果就不会漏掉以“中国科学院”为索引单位元的索引内容。采用这种方法可较少考虑分词的粒度问题。这样不仅能够很好地克服分词颗粒度过大导致相关文档无法检索到的缺点，也使得不必将所有长词都切分成小粒度词，进而导致合并结果数增多，所以这样还能兼顾检索的时间效率。

3.4.4　未登录词识别的准确率要比召回率更重要

我们认为，检索结果中相关文档和不相关文档混杂的情况，还是要比检索不到相关文档要好一些，所以对未登录词识别算法更侧重于准确率，而把召回率放在一个相对次要的地位。

在信息检索中，要尽量保证未登录词识别时不进行错误结合，因为即使有些未登录词没有正确识别出来，但只要没有进行错误地结合，同样可以在最后进行结果合并时检索到；若将它们错误地结合成未登录词，则可能导致无法正确检索到相应文档。

基于以上几个特点，我们提出一种面向大规模中文信息检索的分词算法。其流程如图 3.4 所示。

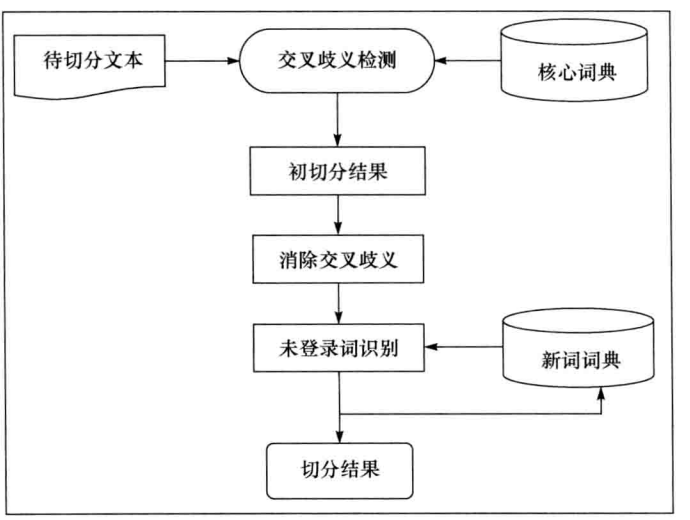


图 3.4　面向大数据的中文信息检索分词算法流程图

首先，根据词典找出句子中所有的交叉歧义。然后，依次利用统计信息进行消歧和未登录词识别。我们将消歧和未登录词识别分成两步处理，主要是因为已知词和未登录词的概率信息有自己的特点，有些适合判断已知词的概率统计方法不适合新词识别，分开来采用不同的方法处理要好一些。

另外，考虑到有些未登录词通常会不止一次出现在文中，所以识别出来的新词都加入到一个新词词典中，便于下一次切分时使用。为了避免错误识别出的新词影响已知词的正确切分，新词词典只用于未登录词识别。

基于以上方法，我们实现了一个原型系统——IRSEG 系统。

3.5 基于双数组 Trie 树优化算法的词典

这里介绍的双数组 Trie 树在进一步提高 Trie 树速度优势的同时，克服了 Trie 树结构在空间上的缺点。

3.5.1 双数组 Trie 树算法介绍及其优化

Trie 树的一个结点相当于一个状态，一条树枝相当于一次状态转移。算法在空间上浪费严重，因为通常的 Trie 树都不是满树。考虑到 Trie 树在本质上是一个有限状态自动机，所以有研究者提出将 Trie 树转换成 3 个线性数组的方法，并在此基础上进行进一步改进将 3 个数组压缩成 2 个(Aoe,1989)，如图 3.5 所示。

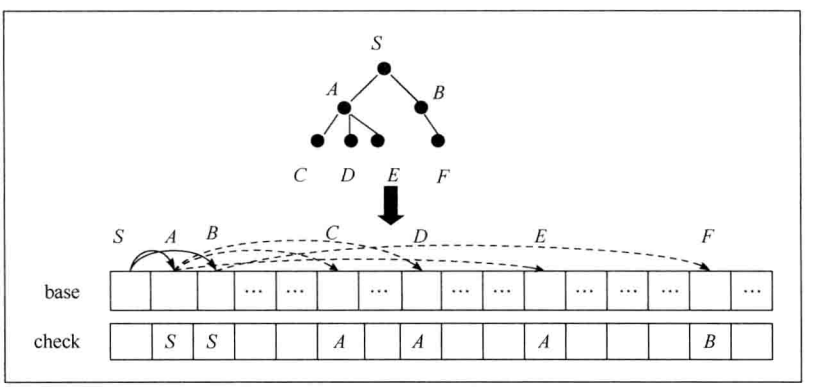


图 3.5 Trie 树到双数组的转换示意图

由图可以看出，Trie 树被扁平化成两个线性数组 base[]和 check[]。base 数组和 check 数组中的元素是一一对应的，base 数组中的每一个元素相当于 Trie 树的一个节点，其值做状态转移的基值，check 值相当于校验值，用于检查该状态是否存在。对于从状态 s 到状态 t 的一个转移，必须满足如下两个条件。

(1) base[s]+c=t。

(2) check[t]=s。

其中，c 是输入变量。

双数组 Trie 树状态转移图如图 3.6 所示。

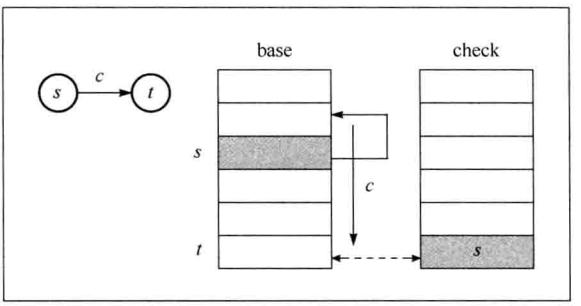


图 3.6　双数组 Trie 树状态转移图

令 i 为数组下标，base[i]和 check[i]均为 0 时表示该位置为空，base[i]为负值时表示该状态为一个可结束状态，两个数组的构造方法如下。

对于状态 AB_1、AB_2、AB_3、…、AB_n，状态 A 在数组中的下标为 i，check[i]=0，令 A 的 base 值 base[i]=k，k 满足条件：base[$k+B_1$]=0，check[$k+B_1$]=0，base[$k+B_2$]=0，check[$k+B_2$]=0，…，base[$k+B_n$]=0，check[$k+B_n$]=0。也就是说，k 的值只要使 A 的直接子结点都能放入数组即可。k 的值确定以后，状态 AB_1，AB_2，AB_3，…，AB_n 在数组中的下标随即确定，分别为 $k+B_1$，$k+B_2$，$k+B_3$，…，$k+B_n$，同时 check[$k+B_1$]=check[$k+B_2$]=…=check[$k+B_n$]=i。

数组构造完成之后，要查找一个关键码 Ab，只需判断 check[base[A]+b]是否等于 A，如果是，则表示 Ab 在 Trie 树中搜索到，否则搜索失败。

虽然将 Trie 树转换成两个数组已经有效地降低了 Trie 树结构的空间浪费，但是观察生成的数组，会发现其中仍然还会存在较大的数据稀疏。为了进一步减少数组空间浪费，我们对构造数组的算法进行优化，在构造数组的时候加入一种排序策略，即每次都先处理当前分支结点最多的结点。

考虑到在双数组 Trie 树算法中，每一个结点在数组中的位置都是由其父亲结点也就是上一状态的 base 值决定。一个结点其 base 值的确定取决于数组的当前空闲位置以及该结点的直接子结点。一个结点的直接子结点越多，该结点在找 base 值时所遇到的冲突也就越多。优先处理分支较多的结点，有利于减少冲突，在处理完分支较多的结点之后，数组中通常会有一些零散空闲位置。这时候处理剩下分支数较少的结点，通常能充分利用这些零散的空闲位置，一般不需要重新增长数组来扩充容量。这样可以避免数组增长过大，减少数据稀疏。

优化后的双数组 Tire 树构造算法如下。

(1) 初始化活动结点列表,把第一层结点加入该列表。

(2) 如果活动结点列表不为空,则在活动结点列表中,选取直接子结点数最多的结点为当前结点,否则算法结束,数组构造完成。

(3) 访问该结点,决定其在 base 数组中的值,确定各直接子结点在数组中的位置,令各直接子结点在 check 数组中的值为当前结点的数组下标。

(4) 将当前结点的直接子结点加入活动结点列表,重复步骤(2)。

Trie 树结构如图 3.7 所示。

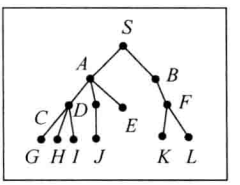


图 3.7 Trie 树结构

构造完成后,查询算法如下。

(1) 读入输入变量 c。

(2) t:=base[s]+c

If check[t]==s then

next state:=t

else fail

endif

(3) 若 base[t]不为负,重复步骤 1;否则,t 为一个可结束状态。

3.5.2 利用优化的双数组 Trie 树算法组织词典

设一个词表为 aa,aab,aad,bc,be,bed,cd,则该词表对应的 Trie 树如图 3.8 所示。

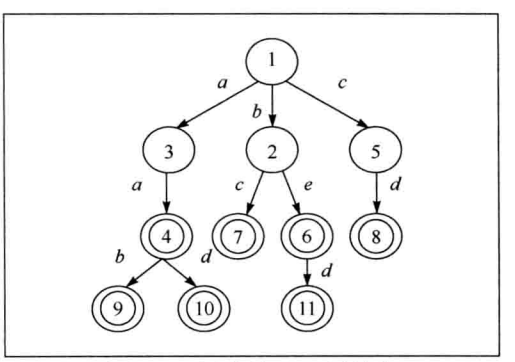


图 3.8 词表的 Trie 树结构

数组初始化均为0，构造数组时，确定结点访问顺序，如结点序号。由于无论英文字母还是汉字在计算机中都由内码唯一表示，所以可用内码一一映射成序列码。假定各字母的序列码为 a-1，b-2，c-3，d-4，e-5。因为词表中所有状态首字母“a，b，c”的序列码分别是1，2，3，所以必须把数组中这三个位置留给首字母。因为叶子结点的表示是用该结点 base 数组下标乘以－1 作为其在 base 数组中的值，而可结束状态表示是该结点 base 数组值乘以－1。为了使二者不产生混淆，所以每个具有可结束状态的非叶子结点其 base 数组值不能等于其数组下标。

访问结点2：该结点状态 b，序列码为2，对应数组下标为2，有两个分支结点，两个转移的变数序列码为3和5。所以，base[2]取1即可，因为 base[1＋3]＝check[1＋3]＝base[1＋5]＝check[1＋5]＝0，令 base[2]＝1，check[4]＝check[6]＝2，结点6和7的数组下标为6和4。

访问结点3：该结点状态为 a，序列码为1，对应数组下标为1，只有一个状态转移的输入变量 a，序列码为1，base[1]可取4，因为 base[4＋1]＝check[4＋1]＝0，令 base[1]＝4，则结点4的数组下标为5，check[5]＝1。

访问结点4：该结点状态为 aa，已知其数组下标为5，由两个分支结点，两个转移的输入变量序列码分别为2和4，base[5]可取6，因为 base[6＋2]＝check[6＋2]＝base[6＋4]＝check[6＋4]＝0，又因为该结点是一个可结束状态，所以令 base[5]＝(－1) * 6，两个分支结点9和10的数组下标分别为|base[5]|＋2＝8和|base[5]|＋4＝10，check[8]＝check[10]＝5。

访问结点5：该结点状态为 c，序列码为3，对应数组下标为3，有一个状态转移的输入变量 d，序列码为4，base[3]可取3，因为 base[3＋4]＝check[3＋4]＝0，令 base[3]＝3，子结点8的数组下标可知为7，check[7]＝3。

访问结点6：该结点状态为 be，已知对应数组下标为6，有一个状态转移的输入变量 d，序列码为4，base[6]可取5，因为 base[5＋4]＝check[5＋4]＝0，又因为该结点是一个可结束状态，所以令 base[6]＝－5，其子结点11的数组下标为9，check[9]＝6，对于其他没有子结点的叶子结点，其 base 值用－1乘以相应的数组下标即可。

base[]＝{4，1，3，－4，－6，－5，－7，－8，－9，－10}

check[]＝{0，0，0，2，1，2，3，5，6，5}

数组构造完成后就可以进行查询了。如果要查字符串 bec 是否词表中的一个词，首先由状态 b 的序列号2得到 base[2]为1，接下来的输入变量是 e，序列号为5，base[2]＋5＝6，check[6]为2，所以 be 是一个状态，可以继续。因为 base[6]为－5，接下来的输入变量 c 序列号为3，|base[6]|＋3＝8，check[8]为5，不是 be 的数组下标6，所以 bec 不是词表中的一个词。如果查的是 bed，d 的序列号为4，|base[6]|＋4＝9，check[9]为6，同时 base[9]等于－9，所以可以判断 bed 是词表

中的一个词。

由上述查询过程可以知道，双数组 Trie 树优化算法的查询只需要进行简单的整数相加即可。查询开销只与查询词长度有关，算法时间复杂度为 $O(n)$，n 为查询词的词长。下面我们通过实验来说明这个算法的空间和时间效率。

3.5.3 实验结果与分析

在前面已经比较过目前常用的几种词典算法，其中速度最快的是双字 Hash 和 Trie 树算法。所以我们将双数组 Trie 树算法和这两种词典算法在相同环境下进行比较。实验硬件环境是 CPU 1.5G，内存 512M，操作系统为 Windows XP，所用词典总共包括 75 784 个词条。三种算法均用 C++实现，一共进行了三项实验。

(1) 分别用未加入优化策略的双数组 Trie 树算法与优化后的双数组 Trie 树算法生成词典的双数组文件，比较空间利用率。

实验结果是未加入优化策略生成的数组，长度为 140 438，加入优化策略后生成的数组长度为 114 624。数组的空间利用率由 85.44%提高到 91.27%。可知，当词典条数越多时，采用优化策略算法提高的空间利用率也会越高。

(2) 比较双字哈希索引机制算法和优化的双数组 Trie 树算法查找词语的速度。为了更接近真实语言环境，我们首先分别用二者把词典中所有的词查一遍，再分别对语料库中出现的所有词进行查询。采用的语料库为已经切分好的《人民日报》1998 年 1 月份语料，大小为 8.04MB。实验结果如图 3.9 所示。

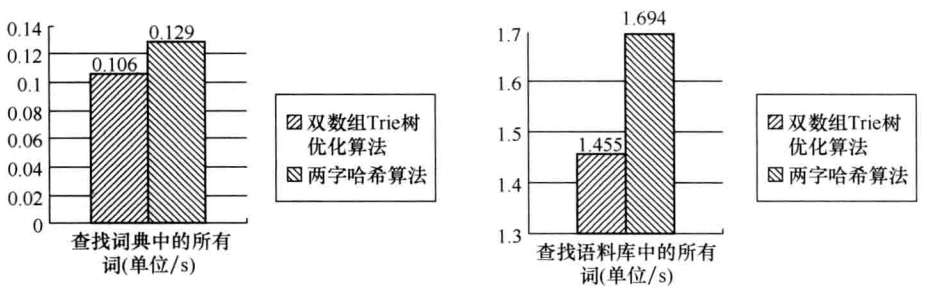


图 3.9 双数组 Trie 树算法和双字 Hash 算法的实验结果比较

从实验结果可以看出，双数组 Trie 树优化算法的查询速度明显要比双字 Hash 算法快。

(3) 比较普通 Trie 树算法和双数组 Trie 树优化算法用于最大匹配自动分词的速度。在这里我们采用正向最大匹配法，语料库文本均为未切分的《人民日报》1998 年语料。实验结果如图 3.10 所示。

文本 1:1998 年 1 月语料,大小为 4 092 478 字节
文本 2:1998 年 2 月语料,大小为 4 153 811 字节
文本 3:1998 年 4 月语料,大小为 4 666 292 字节

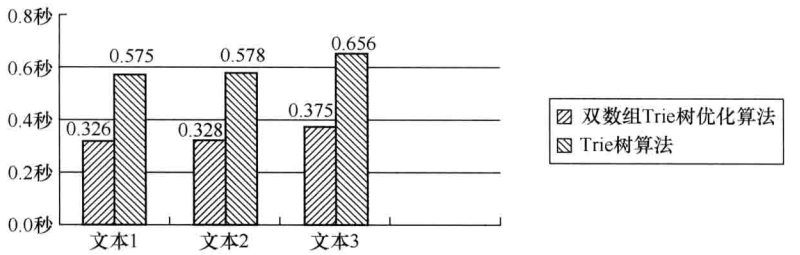


图 3.10　双数组 Trie 树算法和 Trie 树算法的实验结果比较

由上述结果可以得出 Trie 树算法的平均速度为 6.8MB/s,而优化后的双数组 Trie 树算法的平均速度为 12MB/s,几乎比前者高出了一倍。

基于逐字二分的双字 Hash 算法和普通 Trie 树算法的切分速度差不多,所以可以说基于双数组 Trie 树词典的切分速度比目前我们知道的几种常用词典算法都要快。

3.6　本章小结

本章介绍了中文分词技术中的难点以及目前常用的一些切分算法,结合实验结果分析了分词系统对中文信息检索的影响,在此基础上,提出了适用于大数据应用环境下的分词算法特点。考虑到大数据应用环境下用于信息检索的分词算法对速度要求比较高,词典算法采用双数组 Trie 树优化算法。该算法查询一个词语最多只需要进行 $n-1$ 次整数相加,时间复杂度是 $O(n)$,n 为查询词长度。实验显示,无论是单纯的词语查询速度还是最大匹配切分速度,双数组 Trie 树算法都要比目前词典机制中常用的普通 Trie 树算法和双字 Hash 算法快。

相关工作与扩展阅读

对语义复杂的中文进行分词处理,一直是学术界研究的热点。*ACM Transactions on Asian Language Information Processing* 和《中文信息学报》等长期以来对相关研究领域的成果进行了关注。Luck 和 Kwok(2002)介绍了中文信息检索中的单词划分和标记化技术。Peng 等(2002)探索了中文信息检索中的词划分和标记化技术。很久以来,研究者都使用(Rijsbergen,1979)的停用词表。Croft 等(2010)对文本处理中的文本统计、文档解析、文档结构和标记、链接分析、信息抽取

等主题内容进行了探讨。苗夺谦和卫志华(2007)对自动分词、分词词典中有关分词词典的构造、基于词属性的分词词典以及机械分词方法(包括正向最大匹配方法、逆向最大匹配算法、邻近匹配算法、最短路径匹配算法、基于统计的最短路径分词算法)等进行了算法说明。另外,由于人名、地名和其他实体随时可以产生和被命名,所以专有名词不能被词典完全覆盖,对专有名词自身也存在一系列挑战,如指代、消歧、分类等,在这个方面的早期研究之一是 Coates-Stephens(1993)的研究。另外,赵欢和朱红权(2009)对双数组 Trie 树分词算法进行了优化,在采用 Trie 树构造双数组 Trie 树的过程中优先处理分支节点多的结点以减少冲突,构造一个空状态序列,将冲突的结点放入 Hash 表中,不需要重新分配结点,然后利用这些方法构造一个中文分词系统,并与其他几种分词方法进行对比,结果表明优化后的双数组 Trie 树插入速度和空间利用率得到了很大提高,且分词查询效率也得到了提高。黄德根等(2009)提出了基于子词的双层 CRFs 中文分词方法,旨在解决中文分词中切分歧义与未登录词的问题,该方法是建立在基于予词的序列标注模型上,第一层利用基于字 CRFs 模型来识别待测语料中的子词,减少了子词的跨越标记错误和增加了子词识别的精确率;第二层利用 CRFs 模型学习基于子词的序列标注,对第一层的输出进行测试并进而得到分词结果,基于子词的双层 CRFs 模型能够更加有效地利用子词来提高中文分词的精度。吴晶晶等(2009)通过研究目前中文分词领域各类分词机制,注意到中文快速分词机制的关键在于对单双字词的识别,提出了一种快速中文分词机制,即双字词-长词哈希机制,通过提高单双字词的查询效率来实现对中文分词机制的改进。

参 考 文 献

高凯,郭立炜,许云峰.2010. 网络信息检索技术及搜索引擎系统开发. 北京:科学出版社.

黄德根,焦世斗,周惠巍.2009. 基于子词的双层 CRFs 中文分词. 计算机研究与发展,47(5):962-968.

苗夺谦,卫志华. 2007. 中文文本信息处理的原理与应用. 北京:清华大学出版社.

王思力.2006. 面向大规模信息检索的中文分词技术研究. 中国科学院计算技术研究所博士学位论文.

吴晶晶,荆继武,聂晓峰,等.2009. 一种快速中文分词词典机制. 中国科学院研究生院学报,26(5):703-711.

赵欢,朱红权. 2009. 基于双数组 Trie 树中文分词研究. 湖南大学学报,36(5):77-80.

Aoe J. 1989. An efficient digital search algorithm by using a double-array structure. IEEE Transactions on Software Engineering,15(9): 1066-1077.

Coates-Stephens S. 1993. The analysis and acquisition of proper names for the understanding of free text. Computers and Humanities,26:441-456.

Croft W B, Donald M, Trevor S. 2010. Search Engines: Information Retrieval in Practice. New

York: Pearson Education Press.

Luch R W P, Kwok K L. 2002. A comparison of Chinese document indexing strategies and retrieval models. ACM Transactions on Asian Language Information Processing, 1(3): 225-268.

Peng F, Huang X, Schuurmans D, et al. 2002. Investigating the relationship between word segmentation performation and retrieval performance in Chinese IR// Proceedings of the 19 th International Conference on Computational Linguistics: 72-78.

Rijsbergen V. 1979. Information retrieval. London: Butterworths Press.

Schubert F, Li H. 2004. Chinese word segmentation and its effect on information retrieval. Information Processing & Management: 161-190.

第 4 章　基于层次隐马尔可夫模型的浅层词法分析

4.1　概　　述

在大数据文本挖掘中，自然语言处理技术占有重要地位。词法分析中的分词、标注、未登录词识别等浅层词法分析结果，在大数据挖掘中起着基础性的重要作用。在以往的对新信息进行检测的过程中，研究者基本上都采用了自然语言处理的相关技术，常用的处理过程包括词干还原、词性标注、命名实体识别、查询分析等。还有的研究者进一步引入了一些先验的本体语义知识，如 WordNet、同义词典、概念模糊集、语义树等。到目前为止，研究者更多的是关注句子检索方法，尤其是在对新信息的检测建模技术方面，往往是采取简单的语言分析工具。我们的研究和实验表明，前期的语言分析往往能起到更关键的作用。在预处理阶段，英文需要进行断句、词汇切分与词形还原，中文对应的浅层处理过程是汉语分词。浅层语言分析涉及的主要内容如图 4.1 所示。其中，分词起着重要的基础性作用。它既是句法分析等深层处理的基础，也是机器翻译、信息检索和信息抽取等应用的重要基础。

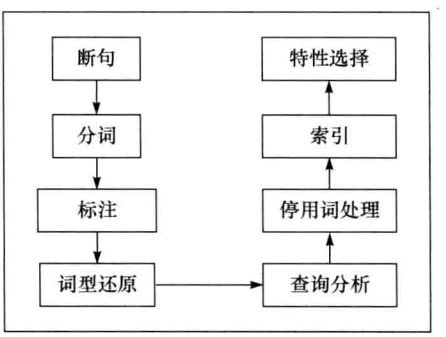


图 4.1　浅层语言分析

本章给出基于层次隐马尔可夫模型的浅层词法分析，旨在将分词、切分排歧、未登录词识别、词性标注等分析任务融合到一个相对统一的理论模型中。在切分排歧方面，提出歧义排除的 N 最短路径策略。在未登录词识别方面，本章提出的方法采取两层隐马尔可夫模型识别出普通的人名地名和复杂嵌套的地名和机构名，计算未登录词的概率，将未登录词的计算结果加入到二元切分词图中，运用基于类的隐马尔可夫模型切分方法，实现未登录词和普通词的统一竞争和筛选。在此理论框架的基础上，研发出汉语词法分析系统的 ICTCLAS。对汉语分词中的

歧义和未登录词处理，本章也给出其他的检测与处理方法。

4.2　英文浅层分析的实现

4.2.1　英文断句与词汇切分

语言处理的第一个环节就是将一个完整的句子从文本中的字符流中切分开来，即所谓的断句。中文的句子标记（如句号、问号、感叹号）很少存在歧义，因此断句相对简单。这里主要介绍英文的处理。

英文的句子主要是以问号（?）、感叹号（!）、引号（'）和句点（.）作为结束标记。在没有上下文的情况下，单个词语或短语也可以作为一个句子。有的句子不带标点，直接用空格或者回车换行来表示句子的结束。回车换行符号仅仅等同于连接的空格，断句时需要根据下一个词语进行区分。问号（?）、感叹号（!）作为句子的结束符号，基本上不存在歧义，但并不是所有的句点（.）都是句子的结束符号，它还可以是多种词汇的组成部分，如缩略名、数字等。在布朗语料库中总共包含了48 885个句子，其中就有 3490 个句子中至少包含了一个作为非终结符号的句点。因此，需要在词汇切分的基础上进行断句，而断句的歧义往往是由词汇切分歧义引起的。

词汇切分的目标就是从原始的字符流中分离出类似于词语的独立单元序列，虽然英语词汇切分的任务类似于汉语的分词，但难度要小得多，关键在于考虑分隔符在词汇中的特殊使用。对于日期、数字、URL 等规则明确的情况，可以采用正则表达式进行处理。对于其他常见的特殊词汇，大部分都收录在一部 15 000 条词条的英文词典中。图 4.2 给出了英文词汇切分与断句的实现算法，实验结果表明该算法能够满足新信息检测的需要。

1. 对一个待分析的字符串（S），如果当前位置 P 已经是 S 的末尾，结束；否则从当前位置 P 开始，由左至右进行扫描，读取一个子串 W 直到空格或者字符串结束符号为止，同时更新 P。

2. 如果 W 在词典 *LEXICON* 中存在，作为词汇单位输出，转至 1；否则

① 如 P 在 S 的末尾，输出 W 并设置句子分隔标记，转至 1。

② 如 W 的最后一个符号属于集合{".","!","?"}；或者 W 的最后一个符号为{""","}","}"}，而且下一个子串 W 的首字母大写，在 W 的最后一个字符前增加词汇分割符号，依次输出这两个词汇单元，并设置句子分隔标记，转至 1。

③ 如果 W 的首字符属于集合{"(","{","[","<",""""}，在 W 的首字符之后增加词汇分割符号，依次输出这两个词汇单元。

④ 否则，采取相关正则表达式进行校验 W，输出 W 这个词汇单元，转至 1。

3. 如果不是字符流尾部，转入 1。

图 4.2　英文词汇切分与断句算法

4.2.2 词性标注

词性标注是自然语言浅层理解的一个重要环节，可以帮助系统自动判定词语所属的语法范畴，为进一步处理提供更高层面的支持。词性标注的主要任务是消除词性兼类歧义，对于新信息检测来说，它的实际意义如下。

(1) 能够在很大程度上消除词义歧义，如对下面的两个句子。

"The pill-induced abortion *can* be painful, causing bleeding and nausea for days"
"Food in *cans* is called *canned* food."

第一个句子中的 can 是情态动词，仅起到了词汇连接作用，并不具备实际语义。第二个句子中的第一个"cans"表示的是名词"罐头"，第二个"canned"表示的是形容词"罐装的"，它们都表达了具体的实体对象。仅从词形出发，系统很难判别它们之间的差别，往往会作为停用词统一处理。

(2) 具有提高句子检索性能的潜力。词性标注可以帮助我们保留表征实际意义的名词、动词、形容词以及数词等实词，同时滤掉对检索不具实际影响的介词、代词、连词与冠词等。相对于纯粹从词形出发的过滤手段，词性更能有效消除噪声，在句子层面的检索上更具有潜力。

(3) 提高区分信息新颖程度的能力。新信息本质上是寻找与以前不同的相关信息，词性标注结果有助于甄别同形异用的词汇，同时还可以帮助对词汇进行语法范畴的分类。

按照学习过程与标注过程的关系，可将词性标注方法分为监督型与非监督型两种。监督型标注器一般都采取事先已经标注好的语料库作为知识来源，从中学习抽取出标注过程需要的资源，如标注词典、词语/词性频率、词性序列的概率以及规则集合。非监督词性标注模型不需要已经标注好的语料库，往往采取比较复杂的计算方法自动地进行词聚类，根据聚类结果来估算相关概率，或者推导出上下文规则。按照采用的技术方法来划分，还可将词性标注分为规则方法、统计方法与神经网络方法。规则往往是语言学家手工整理，有的研究者还尝试自动地从语料库中自动地抽取规则(Brill，1995)，传统的规则方法往往根据上下文信息来标记未知词与歧义词的词性，如规则"det－X－n＝X/adj"说的就是"如果歧义词 X 在限定词与名词之间，则其词性为形容词"，因此"a good boy"中的"good"为形容词，而"I'm telling you this for your good."中的"good"为名词。

除上下文信息之外，构词规则也往往起到关键作用，如字母大小写、前后缀、标点等。典型的系统有 TAGGIT (Greene，1971) 和 Brill Tagger (Brill，1992) 等。统计方法可以是任何一种结合了概率信息的标注模型，其目标是为句子中的词串选择一个最可能的词类序列。20 世纪 80 年代初，英国 Lancaster 大学的研究小组

设计的 CLAWS 系统第一次在词性自动标注中运用了统计语言学的模型。在 CLAWS 之后，词性自动标注多采用统计的方法，典型的统计方法有 n 元语言模型、最大熵方法、隐马尔可夫模型等。实现 n 元模型标注最常用的方法是 Viterbi 算法。20 世纪 80 年代中期后，随着人工神经网络（即连接机制）的兴起，运用神经网络的方法进行词性自动标注的研究也取得了一定的成果。

在英文词性标注中，主要工作是在 Brill 标注器（Brill，1992）的源代码基础上作了进一步改进，采用的词性标注集为宾州树库的标注集合，采用的是基于错误驱动的词性标注方法，其基本过程如下。

（1）初始词性赋值。

（2）对比正确标注的句子，自动学习结构转换规则。

（3）利用转换规则调整初始赋值。

Brill 词性标注器除了存在严重的内存泄漏等技术缺陷之外，不能实际应用的原因还在于它只能处理一个单句的词汇串，而不能适应实际的自由文本。例如，原来的系统只能接受“I＃'m＃a＃Ph. D. ＃candidate＃.”，而不能正确处理实际句子“I'm a Ph. D. candidate.”。为此我们做了如下改进，在词性标注之前，增加了断句与词汇切分预处理过程；针对新信息检测的需要，增加了新的词条以及部分针对主题分析的规则集合。

改进后的词性标注器能较好地适应各种文本，并可以有针对性的分析主题与文档内容。

4.2.3　词干抽取与词形还原

信息检索的关键问题是判断数据集合中哪些文档能够满足用户的信息需求。然而与中文不同的是，文档和查询中的英文词汇往往有多种多样的变种。例如，不进行自然语言处理，computing 与 computation 就不会被看做是同等的。绝大多数情况下，同一词语的各种词形变换形式在语义解释上往往相同，因此从信息检索的角度来看，可以认为它们是无差别的。

在多数文档级信息处理应用中，词干抽取是常用的一种词形变换手段。词干抽取可以处理词形的变化形式并把所有同根词转变为单一形式。例如，computing、computation、compute、computer 以及 computable 等词语统一转换为词干 comput。对于信息检索来说，词干抽取减少了不同词语数量，帮助识别了相似词语，在不进行语言分析的情况下，能以较小的代价提高检索性能。比较有影响力的词干抽取工具有 Porter Stemmer、Paice/Husk Stemmer、Lovins Stemmer 等。

词干抽取基本上能满足大规模文档级信息处理的要求。然而对于句子检索与新信息检测来说，它还远远不能达到要求，其主要缺陷在于。

（1）词干抽取本身就存在欠处理与过处理问题。自然语言本身的结构并不存

在一个绝对规则，因此词干抽取不可避免地会出现错误。一方面，本来应该合二为一的词在词干抽取之后仍然不同(欠处理问题)；另一方面，本来存在很大差别的词对往往在词干抽取之后变成了同形词(过处理问题)。例如，“experiment”与“experience”词干抽取的结果都是“experi”。

(2) 一个句子中包含的词汇数目本身有限，词干抽取的欠处理降低了相关句子被检索到的概率，从而降低了句子检索的召回率。同样，一个词汇的过处理往往会导致非相关句子被错误检索，从而降低句子检索的准确率。

(3) 新信息检测主要是从词汇之间的差异判断信息的新颖程度，而词干抽取更多的是将不同词性、不同词形的词汇融合为同一个词干。这在一定程度上抹杀了信息之间的差别，从而降低了新信息检测的召回率。

基于以上考虑，在新信息检测中，我们采用更为精细的词形还原，而没有采用常用的词干抽取。词形还原的目标是还原词语的词根，去除在实际文本中出现的各种形态，主要包括名词的单复数、动词的各种时态与语态变化、形容词和副词的比较级与最高级形式。与词干抽取不同的是，词性还原的结果是一个实际的词汇，并且与原词的词性保持一致，能在去除形态变换的同时，最大限度地保留原词的意义，主要过程如图 4.3 所示。

输入：词语 W，其词性 P

1. 根据词语 W 的词性 P，查询英文词典 Lexicon，如果存在，输出 W；否则转至 2。
2. 根据词语 W 的词性 P 查找对应的特例列表 ExceptList(P)，如果属于 ExceptList(P)中的特例情况，直接输出对应的词根 R，返回；否则转至 3。
3. 遍历对应词性的转换规则表 TransRule(P)，依次尝试每条转换规则，转换后的结果为 R'，查询 R'在英文词典 Lexicon 中是否存在，如果存在，输出对应词根 R'，返回；否则尝试下一条规则，直到遍历完成为止。
4. 输出 W。

图 4.3 英文词形还原算法

4.3 停用词处理与特征词选择

文档主要是通过其文本内容中的词语相互区分，但不同词语的分辨力千差万别。我们需要过滤掉没有分辨力或者分辨力很低的词语并保留具有一定分辨力的特征词语，提高系统性能，这就需要进行停用词处理与特征选择。

4.3.1 停用词处理

首先人工整理收集一份停用词表，一旦文档中出现了停用词表中的词，则将其删除，最后只保留停用词表之外的词语，具体过程如下。

(1) 保留所有的名词(N)、动词(V)、形容词(J)以及副词(R)。

(2) 在文档标题或者主题描述中出现的数词 CD 予以保留，除此之外的所有词语均予以过滤。

(3) 剩余的结果如果出现在停用词表中，将其过滤。

在很多情况下数词并不影响信息检索结果，很多研究者往往将其视为停用词不予保留，或者不参与相关性计算。但是，在文档标题或者主题描述中出现的数词往往具有很强的区分能力，例如 TREC 2004 年主题 N56 中的主题描述(词性标注结果)。

> "Woodstock/JJ 99/NN music/NN festival/NN reunion/NN in/IN Rome/NNP ,/, NY/NNP"

这里的数词"99"强调的是 Woodstock 音乐节的年份，以示和其他年份的音乐节进行区分，因此我们保留了有区分能力的数词。

4.3.2　特征词选择

停用词处理仅仅是从语言功用的角度筛选出具有实际语义的词语，还不足以选择出对句子检索具有分辨力的特征词语。由于文档往往采用高频的词语来强调需要表达的特征涵义，而在句子中往往是单一的词语，因此在句子检索中需要进一步选择更有区分能力的特征词。为此引入了文本分类过滤中常用的特征选择过程。句子集合的初始特征中可能存在很多噪声，通过特征选择舍弃一些不太重要的词，将有效地消除噪声词语的影响。特征选择具有降低向量空间维数、简化计算、降噪、防止过分拟合等作用。我们采用互信息与 χ^2 统计相结合的思路。

1. 互信息

互信息度量两个对象之间的相关性。我们采用互信息度量特征词语与主题的关联度。特征 w 对于主题 T 的互信息量 $\mathrm{MI}(w,T)$的计算公式为

$$\mathrm{MI}(w,T)=\log\frac{P(w|T)}{P(w)}$$

其中，$P(w|T)$表示的是特征词语 w 在与主题 T 相关的文档集合中的分布概率；$P(w)$表示的是特征词语 w 在所有文档集合中的分布概率，包括与主题 T 相关和不相关的文档集合。

在系统中，将主题 T 相关的文档作为 $P(w|T)$的概率空间，而将所有 50 个主题的文档集作为 $P(w)$的概率空间。对于具有相同条件概率 $P(w|C_i)$的特征，低频特征的 $P(w)$较小，因此互信息比中频特征要高，出现频率差异很大的特征的互信息大小不具有可比性。选择 $\mathrm{MI}(w,T)$较大的特征组成特征空间，对于每一主题

来讲，特征 w 的互信息越大，说明它与该主题的共现概率越大，因此以互信息作为特征提取的评价标准时应选择互信息大的若干个特征。在实验基础上，我们采用的阈值为 0.1。

2. χ^2 统计

χ^2 统计量用于度量特征词语 w 和主题 T 之间的独立性。$\overline{w}$ 表示除 w 以外的其他特征，$\overline{T}$ 表示除 T 以外的其他主题，那么特征 w 和主题 T 的关系有以下四种情况，即(w,T)、$(w,\overline{T})$、$(\overline{w},T)$、$(\overline{w},\overline{T})$，用 A、B、C、D 分别表示这四种情况的文档频次，总的文档数 $N=A+B+C+D$，χ^2 统计量计算公式为

$$\chi^2(w,T)=\frac{N\times(AD-BC)^2}{(A+C)\times(B+D)\times(A+B)\times(C+D)}$$

当特征 w 和主题 T 之间完全独立时，χ^2 统计量为 0。χ^2 统计量和互信息的差别在于它是归一化的统计量，但是它对低频特征的区分效果也不好。在我们的系统中，根据已有的研究结果(Manning，1998)将经验阈值设置为 3.841，即选取 χ^2 统计量大于 3.841 的特征词组成特征空间。在系统中，特征选择的结果是互信息特征空间与 χ^2 统计量特征空间的交集，目的在于相互弥补两种方法之间的不足，选取区分能力更强的特征词。

4.4 基于层次隐马尔可夫模型的汉语浅层分析及其应用

基于层次隐马尔可夫模型的方法旨在将汉语分词、切分排歧、未登录词识别、词性标注等浅层语言分析任务融合到一个相对统一的理论模型中。具体做法如下。

(1) 在预处理的阶段，采取 N 最短路径粗分方法，快速得到能覆盖歧义的最佳 N 个粗切分结果。

(2) 在粗分结果集上，采用底层隐马尔可夫模型识别出普通无嵌套的人名、地名，并依次采取高层隐马尔可夫模型识别出嵌套了人名、地名的复杂地名和机构名。

(3) 将识别出的未登录词以计算出来的概率加入到基于类的切分隐马尔可夫模型中，未登录词与歧义均不作为特例，与普通词一起参与各种候选结果的竞争。

(4) 在全局最优的分词结果上进行词性的隐马尔可夫标注。

该方法已经应用到汉语词法分析系统 ICTCLAS 中，取得了较好的分词和标注效果。ICTCLAS 在 973 专家组机器翻译第二阶段的评测和 SIGHAN 举办的第一届汉语分词大赛中，取得了不俗的成绩，是目前最好的汉语词法分析系统之一。

4.4.1　层次隐马尔可夫模型

隐马尔可夫模型(Lawrence,1989)是经典的描述随机过程的统计方法,在自然语言处理中得到了广泛的应用。相对于复杂的自然语言现象来说,传统的隐马尔可夫模型仍然略显简单,文献(Shai,1998)对隐马尔可夫模型进行了扩展和泛化并提出了层次隐马尔可夫模型。在文献(Shai,1998)的工作基础上,我们将层次隐马尔可夫模型形式化为六元组,即

$$M=<\Omega_X,\Omega_O,A,B,\Pi,D>$$

其中,Ω_X为有限状态集合;Ω_O为观测结果有限集;A 为状态转移矩阵;B 为状态到观测值的概率矩阵;Π 为初始状态分布;D 为 M 的深度。

层次隐马尔可夫模型与 HMM 的主要区别如下。

(1) Ω_X中状态的表示为 $q_i^d(d\in\{1,2,\cdots,D\})$,其中 i 表示的是该状态在当前层 HMM 状态中的编号,d 是该状态在 M 中的层次深度,所有状态形成了一个深度为$(D-1)$的树型结构,根的深度为 1,最深的叶子深度为 D,$d<D$ 的状态称为内部状态。

(2) 每一个内部状态 $q^d(d\in\{1,2,\cdots,D-1\})$存在子状态,子状态数记为 $|q^d|$,所有子状态构成一个隐马尔可夫链,$(d+1)$层的状态输出可视为该层 HMM 的状态序列。该层 HMM 的状态转移矩阵为 $A(q^d)=(a_{ij}(q^d))$,其中 $a_{ij}(q^d)=P(q_j^{d+1}|q_i^{d+1},q^d)$。同时,各个状态的初始分布为 $\Pi(q^d)=\pi^d(q_i^{d+1})=P(q_i^{d+1}|q_i^d)$,其物理意义可以理解为第 d 层隐马尔可夫模型的某一内部状态 q^d 到第 $d+1$ 层隐马尔可夫模型的激活概率。

(3) 在各层隐马尔可夫模型中,只有第 D 层才有真正能观测到的终结符,即状态到观测的输出概率 $B(q^D)=(b_k(q^D))$,其中 $b_k(q^D)=P(o_k|q^D)$,o_k是观测值,属于一个有限终结符集合。

因此,层次隐马尔可夫模型的参数集合可以表示为

$$\begin{aligned}\lambda &= \{\lambda(q^d)\}_{d\in\{1,2,\cdots,D\}}\\ &= \{\{A(q^d)\}_{d\in\{1,2,\cdots,D-1\}},\{\prod(q^d)\}_{d\in\{1,2,\cdots,D-1\}},\{B(q^D)\}\}\end{aligned}$$

显然,层次隐马尔可夫模型在 $D=1$ 时就会退化成简单的隐马尔可夫模型。

针对汉语浅层语言分析各个层面的处理对象及问题特点,我们引入层次隐马尔可夫模型统一建模,给出基于层次隐马尔可夫模型的汉语浅层语言分析框架。该模型包含原子切分、普通未登录词识别、嵌套的复杂未登录词识别、基于类的隐马尔可夫切分、词类标注等五个层面的隐马尔可夫模型,如图 4.4 所示。其主要部分的功能如下。

(1) N 最短路径粗切分:可快速产生 N 个最好的粗切分结果,粗切分结果集能覆盖尽可能多的歧义。在整个词法分析架构中,二元切分词图是关键的中间数据结构,它将未登录词识别、排歧、分词等过程有机地进行了融合。

(2) 原子切分:词法分析的预处理过程,其任务是将原始字符串切分为分词原子序列。分词原子指的是分词的最小处理单元,分词原子包括单个汉字、标点以及由单字节、字符、数字等组成的非汉字串。例如,"2002.9,ICTCLAS 的自由源码开始发布"对应的分词原子序列为"2002.9/,/ICTCLAS/的/自/由/源/码/开/始/发/布/"。在这层隐马尔可夫模型中,终结符是书面语中所有的字符,状态集合为分词原子,模型的训练和求解都比较简单,不再赘述。

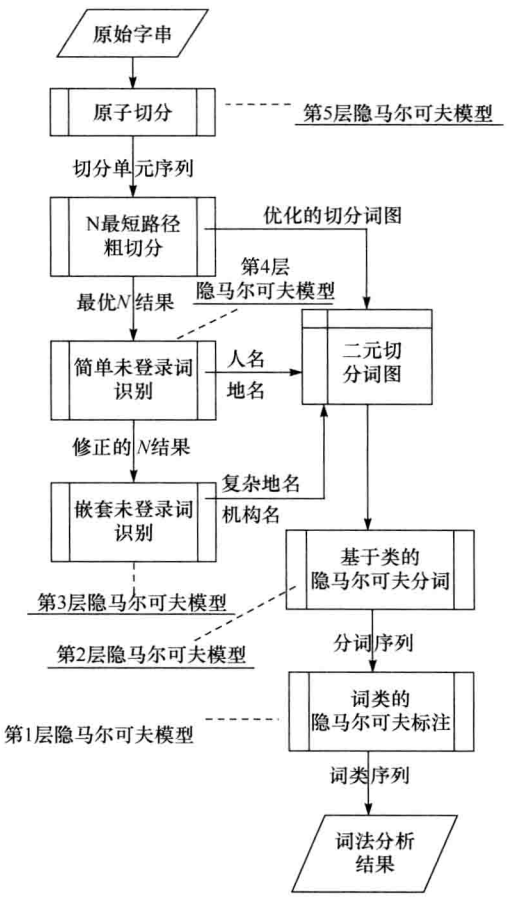


图 4.4 基于层次隐马尔可夫模型的汉语词法分析框架

4.4.2　基于类的隐马尔可夫分词算法

算法处于层次隐马尔可夫模型的第二层，也就是在所有的未登录词识别完成后进行。首先，可以把所有的词按图 4.5 分类。核心词典中已有的每个词对应的类就是该词本身。假定核心词典中收入的词数为 |Dict|，则定义的词类总数为 |Dict| +6。

$$c_i=\begin{cases} w_i, & w_i\text{在核心词典中收录} \\ \text{PER}, & w_i\text{是人名 and } w_i\text{是未登录词} \\ \text{LOC}, & w_i\text{是地名 and } w_i\text{是未登录词} \\ \text{ORG}, & w_i\text{是机构名 and } w_i\text{是未登录词} \\ \text{NUM}, & w_i\text{是数词 and } w_i\text{是未登录词} \\ \text{TIME}, & w_i\text{是时间词 and } w_i\text{是未登录词} \\ \text{BEG}, & w_i\text{是句子的开始标记} \\ \text{END}, & w_i\text{是句子的结束标记} \\ \text{OTHER}, & \text{其他} \end{cases}$$

图 4.5　词的分类

给定一个分词原子序列 S，S 的某个可能的分词结果记为 $W=(w_1,w_2,\cdots,w_n)$，W 对应的类别序列记为 $C=(c_1,c_2,\cdots,c_n)$，同时取概率最大的分词结果 $W^{\#}$ 作为最终的分词结果，则

$$W^{\#}=\arg\max_{W} P(W)$$

利用贝叶斯公式进行展开得到，即

$$W^{\#}=\arg\max_{W} P(W|C)P(C)$$

将词类看作状态，词语作为观测值，利用一阶 HMM 展开得

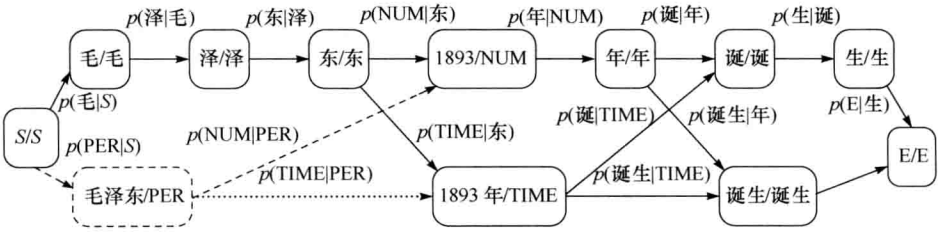
$$W^{\#}=\arg\max_{W}\prod_{i=1}^{n} p(w_i \mid c_i)p(c_i \mid c_{i-1})$$

其中，c_0 为句子的开始标记 BEG，下同。

为计算方便，常用负对数来运算，则有 $W^{\#}=\arg\min_{W}\sum_{i=1}^{n}[-\ln p(w_i \mid c_i)-\ln p(c_i \mid c_{i-1})]$。

根据图 4.5 中 c_i 的定义，如果 w_i 在核心词典收录，可得 $c_i=w_i$，故 $p(w_i \mid c_i)=1$。NUM 和 TIME 两类词的构成符合正则文法，ICTCLAS 可以采用确定性的有限状态自动机进行识别，基本不存在歧义组合的问题，将之视为确定性的词类，和标点符号一样处理，不作为未登录词对待，将相应的 $p(w_i \mid c_i)$ 作为一个常数，从而将问题进一步简化。因此，在分词过程中，只需要考虑未登录词的 $p(w_i \mid c_i)$。在图 4.6 中给出了“毛泽东 1893 年诞生”的二元切分词图。最终所求的分词结果就是从初始节点 S 到结束节点 E 的最短路径。这是个典型的最短路径问题，

可采取贪心算法，如 Dijkstra 算法快速求解。在实际应用基于类的分词时，切分歧义能否在这一模型内进行融合并排解是一个难题。另外一个关键问题在于如何确定未登录词 w_i、识别其类别 c_i，并计算出可信的 $p'(w_i|c_i)$。



说明：

1. 节点中表示的是“词语/类”(w_i/c_i)，节点的权值为类到词语的概率 $p(w_i|c_i)$。
2. 有向边的权值为相邻类的转移概率 $p(c_i|c_{i-1})$；S 为初始节点；E 为结束节点。
3. “毛泽东/PER”相关的虚线部分是人名识别 HMM 作用过之后产生的。

图 4.6 基于类的二元切分词图示例

4.4.3 N 最短路径的切分排歧策略

从构成形态上划分，切分歧义一般分为交叉歧义和组合歧义。“结合/成/分子/时”是个典型的交叉歧义，“这/个/人/手/上/有/痣”中的“人/手”构成了一个组合歧义字段。

从排歧的角度看，切分歧义可以分为全局歧义和局部歧义。全局歧义指的是必须结合当前句子的上下文才能准确排除的歧义。例如，“乒乓球拍卖完了”，在缺乏语境的情况下，可以合理地切分为“乒乓球/拍卖/完/了”和“乒乓球拍/卖/完/了”。与此相反，局部歧义完全可以在句子内部进行排除。根据对大规模语料的统计发现，局部歧义占绝大多数，全局歧义几乎可以忽略不计。

我们采取的是 N 最短路径的切分排歧策略(张华平，2002)，基本思想是在初始阶段保留切分概率 $P(W)$最大的 N 个结果，作为分词结果的候选集合。在未登录词识别、词性标注等词法分析之后，再通过最终的评价函数，计算出真正的最优结果。实际上，N 最短路径方法是最少切分方法和全切分的泛化和综合。它一方面避免了最少切分方法大量舍弃正确结果的可能，另一方面又大大解决了全切分搜索空间过大、运行效率差的弊端。该方法通过保留少量大概率的粗分结果，可以最大限度地保留歧义字段和未登录词。常用的切分算法往往过于武断，过早地在初始阶段做出是否切分的判断，只保留一个自己认为最优的结果，而这一结果往往会因为存在歧义或未登录词而出错，这时后期补救措施往往费时费力，效果也不会很好。表 4.1 给出了 8 最短路径与常用算法在切分结果包容歧义方面的对比测试结果。

表 4.1　N 最短路径与常用算法对比

方法	切分最大数	切分平均数	正确切分覆盖率/%
最大匹配	1	1	85.46
最少切分	1	1	91.80
最大概率	1	1	93.50
全切分	>3,424,507	>391.79	100.00
8 最短路径	8	5.82	99.92

说明：① 切分最大数指的是句子可能的最大切分结果数。

② 切分平均数指的是单个句子平均的切分结果数。

③ 正确切分覆盖率＝正确切分被覆盖的句子数/句子总数。

④ 测试语料大小为 200 万汉字。

此外，我们对最终选择出的唯一切分标注结果进行了开放歧义测试，测试集合是北京大学计算语言学研究所收集的 120 对常见组合歧义、99 对常见交叉歧义，最终组合歧义和交叉歧义排除的成功率分别为 80.00%和 92.93%。

4.4.4　未登录词的隐马尔可夫识别方法

未登录词识别的任务包括确定未登录词 w_i 的边界和类别 c_i，计算 $p(w_i|c_i)$。我们在 N 个候选切分结果的词类序列基础上，引入了高层隐马尔可夫模型来实现未登录词的识别。

1. 未登录词识别角色表

与基于类的隐马尔可夫分词模型类似，我们对初始切分得到的各个词按照其在未登录词识别中的作用进行分类，并将词所起的不同作用称为角色。表 4.2 是人名识别的角色表。与隐马尔可夫分词中定义的类相比（类和词是一对多的关系），角色与词是多对多的关系，即一个词可以充当多个角色，而一个角色也可以对应多个词。

表 4.2　人名识别角色表

角色	意义	示例
A	人名的上文	又/来到/于/洪/洋/的/家
B	人名的下文	新华社/记者/黄/文/摄
C	中国人名的姓	张/华/平/先生；欧阳/修
D	双名的首字	张/华/平/先生
E	双名的末字	张/华/平/先生
F	单名	张/浩

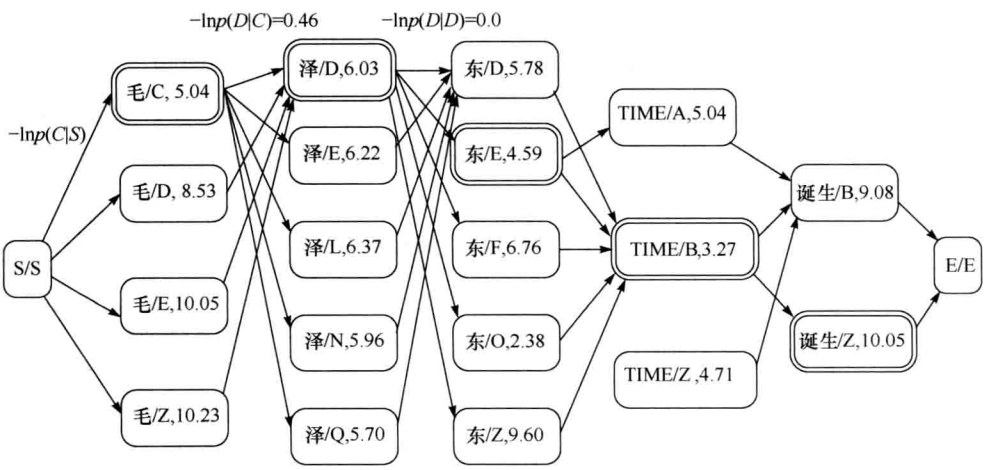
续表

角色	意义	示例
G	人名的前缀	老/刘、小/李
H	人名的后缀	王/总、刘/老、肖/氏
L	译名的首部	蒙/帕/蒂/·/梅/拉/费
M	译名的中部	蒙/帕/蒂/·/梅/拉/费
N	译名的末部	蒙/帕/蒂/·/梅/拉/费
O	日本人名末部	小泉/纯/一/郎
X	连接词	邵/钧/林/和/稽/道/青/说
Z	其他	人民/深切/缅怀/邓/小/平

2. 角色标注与未登录词识别

对于一个给定的初始切分结果 $W=(w_1,w_2,\cdots,w_n)$，在一个角色集合的范畴内，假定 $R=(r_1,r_2,\cdots,r_n)$ 为 C 的某个角色序列，取概率最大的角色序列 $R^{\#}$ 作为最终的角色标注结果。与隐马尔可夫分词的推导过程类似，最终可得到 $R^{\#}=\underset{R}{\operatorname{argmin}}\sum_{i=1}^{n}[-\ln p'(w_i\mid r_i)-\ln p(r_i\mid r_{i-1})]$，其中 r_0 为句子的开始标记 BEG。

$R^{\#}$ 可以通过 Viterbi 算法选优得到(Viterbi,1967;Lawrence,1989)。图 4.7 给出了在词类序列“毛/泽/东/TIME/诞生”的 Viterbi 算法标注人名角色的过程。这里 TIME 是在原子切分阶段通过简单的有限状态自动机识别出来的。



说明：① 图中结点的格式为：词类 c_i/角色 r_i，$-\log P(c_i\mid r_i)$，双线节点 Viterbi 选优结果。

② 图中的有向边权值为相邻角色的转移概率 $-\log P(r_i\mid r_{i-1})$，这里没有全部列出。

图 4.7 角色标注的 Viterbi 算法选优过程

在最大概率角色序列的基础上，可以简单地通过模板匹配实现特定类型未登录词的识别。在图 4.7 中，可以求解出最优的角色标注结果“毛/C 泽/D 东/E TIME/B 诞生/Z”，而 CDE 正好构成一个典型的汉语人名，因此“毛泽东”被识别为人名 PER。识别出来的未登录词为 w，类别为 c，利用隐马尔可夫过程有

$$p(w \mid c) = \prod_{j=0}^{k} p(w_{p+j} \mid r_{p+j}) p(r_{p+j} \mid r_{p+j-1})$$

其中，w_i 由第 p，$p+1$，…，$p+k-1$ 个初始切分单元组成。

最后，识别结果及其概率加入到二元隐马尔可夫模型切分图中，和普通词一样处理，竞争出最佳结果。

3. 未登录词识别

复杂地名和机构名往往嵌套了普通无嵌套的人名、地名等未登录词，如“张自忠路”、“周恩来和邓颖超纪念馆”。对于这种嵌套的未登录词，我们的做法是在低层的隐马尔可夫模型识别过程中，先识别出普通不嵌套的未登录词，然后在此基础上通过相同的方法采取高层隐马尔可夫模型，通过角色标注计算出最优的角色序列，在此基础上进一步识别出嵌套的未登录词。以切分序列片段“周/恩/来/和/邓/颖/超/纪念馆”为例，先识别出“周恩来”和“邓颖超”为人名 PER，得到新的词类序列“PER/和/PER/纪念馆”，最终就可以识别出该片段为机构名。这样的处理优点在于能够利用已经分析的结果，并降低数据的稀疏程度。用来训练隐马尔可夫模型角色参数的语料库是在北京大学计算语言学研究所切分标注语料库的基础上，在甄别出各种类型的未登录词之后，自动转换得到的。

4.5　汉语词法分析系统 ICTCLAS 性能实验与分析

采取层次隐马尔可夫模型的方法，我们研制了汉语词法分析系统 ICTCLAS。下面给出 ICTCLAS 在不同条件下的测试结果，提供 ICTCLAS 在国家 973 英汉机器翻译第二阶段评测中开放测试的结果，并介绍在第一届国际分词大赛中的比赛情况。按照惯例，评测指标包括切分正确率 SEG、上位词性标对率 TAG1、下位词性标对率 TAG2、命名实体(NE)识别的准确率(P)和召回率(R)以及 F 值。

(1) SEG＝切分正确的词数/总词数×100%。

(2) TAG1＝上位词性标注正确数/总词数×100%。

(3) TAG2＝下位词性标注正确数/总词数×100%。

(4) P＝正确识别该类 NE 数/识别出该类 NE 总数×100%。

(5) R＝正确识别该类 NE 数/该类 NE 总数×100%。

(6) $F=\dfrac{R\times P\times(1+\beta^2)}{R+P\times\beta^2}$，这里我们取 $\beta=1$，称为 F-1 值。

4.5.1 词法分析与层次隐马尔可夫模型

我们使用北京大学计算语言学研究所加工的《人民日报》语料库进行了训练和测试。在《人民日报》1998 年一月份共计 1 108 049 词的新闻语料库上进行了如下四种条件下的性能测试。

(1) BASE：基准测试，即仅做隐马尔可夫分词和词性标注，不引入其他层面的隐马尔可夫模型。

(2) +PER：在 BASE 的基础上引入人名识别隐马尔可夫模型。

(3) +LOC：在+PER 的基础上引入地名识别隐马尔可夫模型。

(4) +ORG：在+LOC 的基础上引入机构名识别隐马尔可夫模型。

图 4.8 给出了四种条件下词法分析的分词正确率 SEG、上位词性标对率 TAG1、下位词性标对率 TAG2、人名识别的 F-1 值 FP、地名识别的 F-1 值 FL 以及机构名识别的 F-1 值 FO。

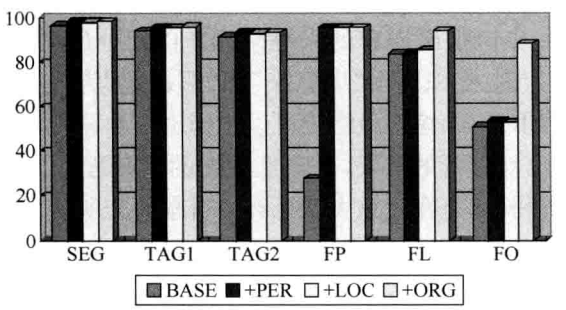


图 4.8 四种条件下的词法分析的性能指标

(1) 随着各层隐马尔可夫模型的加入，词法分析的效果逐步提升。其中，人名识别引入后，切分正确率 SEG 在 96.55%的基础上，增加到 97.96%，增幅最大。人名、地名、机构名等识别隐马尔可夫模型均加入后，切分正确率 SEG、上位词性标对率 TAG1、下位词性标对率 TAG2 分别达到了 98.38%、95.76%、93.52%。这表明各层隐马尔可夫模型对最终词法分析均发挥了积极作用。

(2) 随着各层隐马尔可夫模型的加入，不仅提高了本层隐马尔可夫模型的最终性能，还改进了低层隐马尔可夫模型处理精度。人名识别隐马尔可夫模型加入后，人名识别的 F-1 值从 27.86%提升到 95.40%，低层的分词隐马尔可夫模型的正确率提高了 1.41%；机构名识别隐马尔可夫模型引入后，机构名识别的 F-1 值提高了 35.59%，同时低层的地名识别也提高了 8.49%，人名识别的 F-1 值也达到了最高点 95.58%。其原因在于高层隐马尔可夫模型的成功应用在解决当前问题

的同时也消除了低层隐马尔可夫模型部分的歧义，排除了低层隐马尔可夫模型的错误结果。例如，在人名识别隐马尔可夫模型中很容易将“刘庄的民风很纯朴”中的“刘庄”错误识别为人名，然而高层的地名识别隐马尔可夫模型会正确地将“刘庄”作为地名召回，因此达到了排歧的作用。

4.5.2　ICTCLAS 在 973 评测中的测试结果

ICTCLAS 参加了国家 973 英汉机器翻译第二阶段的开放评测，测试结果如表 4.3 所示(数据来源为国家 973 英汉机器翻译第二阶段评测的评测总结报告)。

表 4.3　ICTCLAS 在 973 评测中的测试结果

领域	词数	SEG/%	TAG1/%	RTAG/%
体育	33 348	97.01	86.77	89.31
国际	59 683	97.51	88.55	90.78
文艺	20 524	96.40	87.47	90.59
法制	14 668	98.44	85.26	86.59
理论	55 225	98.12	87.29	88.91
经济	24 765	97.80	86.25	88.16
总计	208 213	97.58	87.32	89.42

其中，标注相对正确率 RTAG＝TAG1/SEG * 100%，因为采取的词性标注集和 973 专家组的标注集有较大出入，所以词性标注的正确率不具可比性。专家组的开放评测结果表明，基于层次隐马尔可夫模型的 ICTCLAS 能实际的解决汉语浅层语言分析问题，与兄弟单位的类似系统对比，ICTCLAS 的分词结果表现出色。

4.5.3　第一届国际分词大赛的评测结果

为了比较和评价不同方法和系统的性能，第 41 届国际计算语言联合会(41st Annual Meeting of the Association for Computational Linguistics，ACL)下设的汉语特别兴趣研究组(ACL Special Interest Group on Chinese Language Processing，SIGHAN)第一届国际汉语分词评测大赛。报名参赛的分别是来自于中国、美国等 6 个国家和地区，共计 19 家研究机构，最终提交结果的是 12 家参赛队伍。大赛采取大规模语料库测试，进行综合打分的方法，语料库和标准分别来自北京大学(简体版)、宾州树库(简体版)、香港城市大学(繁体版)。每家标准分两个任务(Track)：受限训练任务(Close Track)和非受限训练任务(Open Track)。

ICTCLAS 分别参加了简体的所有四项任务，以及繁体的受限训练任务。其中在宾州树库受限训练任务中综合得分 0.881，名列第一；北京大学受限训练任务中综合得分 0.951，名列第一；北京大学受限训练任务中综合得分 0.953，名列第

二。值得注意的是，我们在短短的两天之内，采取 ICTCLAS 简体版的内核代码，将层次隐马尔可夫模型推广到繁体分词当中，同样取得了 0.938 的综合得分。

4.6 基于单字位置成词概率识别未登录词的算法

4.6.1 字的位置成词概率

在经过歧义处理后，得到的切分片段中会包含一些被切分开来的两个或两个以上的连续单字，我们将这些连续单字称为单字碎片。未登录词识别的目标之一就是将这些碎片中的未登录词正确合并起来。虽然有些在前面识别的已知词也可能是未登录词的一个部分，如“刘海洋”、“平安乡”这样的情况，但是因为算法对于人名的切分标准就是将姓和名切分开来的，所以一般不会影响人名的正确识别。对于地名和机构名等，是要将“平安”这样的名字和后面的特征用字结合在一起的，但是如果名字单独作为一个已知词识别出来，虽然会影响识别和切分的正确率，但是不会导致最终的检索错误。因此，在这里我们未登录词识别的算法只处理单字碎片。

通过观察可以发现，有些汉字在语言中出现的位置是有一定的规律的，如有些字出现的位置大多在词首，如“阿”、“逮”、“尴”；有些多在词尾出现，如“定”、“峰”、“尬”；有些大多是单字出现，不与其他字成词，如“啊”、“很”；有些字则很少单独一个字作为词出现，如“苍”、“究”等。

在未登录词识别阶段，因为已知词都已经经过了处理，切分出来的单字碎片只有两种可能，一种是真正的单字，如“它”、“是”、“的”、“要”之类的词，这些字单字成词概率比较高，虽然它们也与其他字成词，但是一般都是已知词，很少出现在未登录词中。另一种就是我们要识别出来的未登录词，通过我们的观察，这些字通常单独成词的概率都不高。例如，“历时\5 年\的**阿\尔\巴\尼\亚**\高息\集资\10 亿\美元\事件\引发\的\金融\危机”，“**栾\老\寨\人**\的\决心\更\坚硬”，“丽\华\快餐\”。

句子中粗体标示出来的都是要识别成未登录词的单字，经过统计发现，这些连续出现的单字其单独成词的概率都很低。也就是说，这些单字从概率上来说应该是属于一个词的，而非单独成词。另外一些单字，如“人\的”这两个字单字成词的概率都很高，所以不能合并为未登录词。下面引入字的位置成词概率。有些文献把这个概率分为四类，词头(B)、词中(I)、词尾(E)和单字成词(S)。这样字 ci 的位置成词概率就分为以下四类。

P(ci,S)：ci 单独成词的概率

P(ci,B)：ci 出现在词首的概率

P(ci,I)：ci 出现在词中的概率

P(ci,E)：ci 出现在词尾的概

其中，$P(c_i,\text{pos})=\dfrac{N(c_i,\text{pos})}{N(c_i)}$；pos 是指该字出现位置所属的类别；$N(c_i,\text{pos})$是该字出现在位置 pos 的次数；$N(c_i)$是该字在语料库中出现的总次数。

我们定义单字合并的规则如表 4.4 所示。

表 4.4　基于位置成词概率的单字合并规则

合并类型	合并条件
$/c_i/c_{i+1}/\rightarrow/c_ic_{i+1}/$	$(P(c_i,S)+P(c_{i+1},S))/2<s\&\&(P(c_i,B)+P(c_{i+1},I)+P(c_{i+1},E))/3>s$
$/c_1c_2\cdots c_i/c_{i+1}/\rightarrow$ $/c_1c_2\cdots c_ic_{i+1}/$	$(P(c_i,S)+P(c_{i+1},S))/2<s\&\&(P(c_i,I)+P(c_{i+1},I)+P(c_{i+1},E))/3>s$

其中，s 是设定的经验阈值。

考察实验结果，这个值取 0.13 时测试得到的识别准确率最高。

下面是测试识别的几个例子，如表 4.5 和表 4.6 所示。

表 4.5　“德班所在的南非夸祖鲁-纳塔尔省省长恩古巴尼”的未登录词识别

待切分句子	初切分结果	碎片	位置概率	识别结果
德班所在的南非夸祖鲁—纳塔尔省省长恩古巴尼	德/班/所　在/的/南　非/夸/祖/鲁/—/纳/塔/尔/省/省长/恩/古/巴/尼	德/班/	P(德,S)=0.053,P(班,S)=0.07 P(德,B)=0.38,P(班,I)=0.35 P(班,E)=0.37	德班
		夸/祖/鲁	P(夸,S)=0.16,P(祖,S)=0.007, P(鲁,S)=0.10 P(夸,B)=0.58,P(祖,I)=0.034, P(祖,E)=0.068 P(鲁,I)=0.42,P(鲁,E)=0.096	夸祖鲁
		纳/塔/尔/省	P(纳,S)=0.01,P(塔,S)=0.05, P(尔,S)=0.001, P(省,S)=0.16 P(纳,B)=0.35,P(塔,I)=0.44, P(塔,E)=0.24,P(尔,I)=0.67, P(尔,E)=0.32,P(省,I)=0.004, P(省,E)=0.59	纳塔尔省
		恩/古/巴/尼	P(恩,S)=0.01,P(古,S)=0.11, P(巴,S)=0.14 P(尼,S)=0.16 P(恩,B)=0.85,P(古,I)=0.13, P(古,E)=0.22 P(巴,I)=0.18,P(巴,E)=0.09 P(尼,I)=0.59,P(尼,E)=0.29	恩古巴尼

表 4.6 “刘明田朱思雄”的未登录词识别

待切分句子	初切分结果	碎片	位置概率	识别结果
刘明田 朱思雄	刘/明/田/ 朱/思/雄	刘/明/田/ 朱/思/雄	P(刘,S)=0.96,P(明,S)=0.025 P(田,S)=0.2, P(明,B)=0.32,P(田,I)=0.08 P(田,E)=0.52 P(朱,S)=0.93,P(思,S)=0.015 P(雄,S)=0.03, P(思,B)=0.65,P(雄,I)=0.056 P(雄,E)=0.53	刘/明田/ 朱/思雄

当然,为了保证准确率,阈值 s 定的比较低,所以如果遇到一些常见字组成的未登录词,则很难识别出来。例如,“超女”,P(超,S)=0.07,P(女,S)=0.3,二者的平均值大于 0.13,所以会被切分开来。同样,“王大中”中的“大”和“中”也会被切分开;“金泳三”会被切分成“金泳/三”。为了弥补该不足之处,我们引入局部二元串频概率来进行改进。

4.6.2 局部二元串频统计

我们之前用到的统计概率信息都是预先通过语料库训练得到的。在未登录词识别中,因为语料库受规模和时间的限制,必然会有很多未登录词没有在语料库中出现。但是,通常一篇文档中出现的某个未登录词,很多时候都会不止一次的出现在该文档中。在对《人民日报》1998 年 1 月份的标注语料库进行统计后发现,在 19 000多个未登录词中,只有 7000 多个词出现一次,其他 12 000 多个词重复出现的平均次数达到了 5 次。也就是说,有大约 63.3%的未登录词是会在文档中重复出现的。所以,我们希望能通过预先统计待切分文档中二元字对的共现次数,来提高未登录词识别的准确率和召回率。之前对语料库统计得到的概率,相当于一个全局概率。这里对待切分文档统计得到的串频,相当于一个局部概率,所以在这里称之为局部串频。为了避免将一些常见的连续出现的单字也错误结合在一起,如“把这”“说的”这样的情况,局部二元串频不能单独使用,必须结合字的位置成词概率来进行判断。加入局部串频之后的碎片合并规则修改如表 4.7 所示。

表 4.7　基于位置概率信息和局部二元串频的结合规则

合并类型	合并条件
$/c_i/c_{i+1}/\rightarrow/c_ic_{i+1}/$	$(P(c_i,S)+P(c_{i+1},S))/2<s\&\&(P(c_i,B)+P(c_{i+1},I)+P(c_{i+1},E))/3>s\|\|C(c_ic_{i+1})>m\&\&(P(c_i,S)+P(c_{i+1},S))/2<s'\&\&(P(c_i,B)+P(c_{i+1},I)+P(c_{i+1},E))/3>s$
$/c_1c_2\cdots c_i/c_{i+1}/\rightarrow/c_1c_2\cdots c_ic_{i+1}/$	$(P(c_i,S)+P(c_{i+1},S))/2<s\&\&(P(c_i,I)+P(c_{i+1},I)+P(c_{i+1},E))/3>s\|\|C(c_ic_{i+1})>m\&\&(P(c_i,S)+P(c_{i+1},S))/2<s'\&\&(P(c_i,B)+P(c_{i+1},I)+P(c_{i+1},E))/3>s$

其中，$C(c_ic_{i+1})$是字串 c_ic_{i+1}在待切分文档中出现的次数；s'和 m 是人工设定的阈值。也就是说，如果字串(c_ic_{i+1})在被切分文档中重复出现次数超过某个数时，$(P(c_i,S)+P(c_{i+1},S))/2$ 的阈值可以增大为 s'；s'为经验值，在实验中取 2.4。

4.6.3　有关未登录词识别的实验结果

这部分实验环境用已经切分过的《人民日报》1998 年 1～6 月语料库做训练集，用《人民日报》1998 年 1 月的语料库做封闭测试，用 SIGHAN05 的部分测试集做开放测试。通常采用的未登录词识别的评测指标为

$$\text{准确率 } P=\frac{N(\text{识别正确的未登录词})}{N(\text{识别的未登录词})}$$

$$\text{召回率 } R=\frac{N(\text{识别正确的未登录词})}{N(\text{文本中正确的未登录词})}$$

我们的实验分为两组。

(1) 根据字的位置成词概率识别未登录词。

(2) 用字的位置成词概率结合局部二元串频统计识别未登录词。

实验结果如表 4.8 所示。

表 4.8　未登录词识别实验结果

语料	实验 1(P/R)	实验 2(P/R)
《人民日报》1998 年 1 月语料	0.741/0.681	0.815/0.762
北京大学语料	0.732/0.680	0.801/0.729
微软研究院语料	0.310/0.213	0.361/0.292

可以看出在加入串频统计之后，未登录词识别的 P 值和 R 值都有所提高。实验结果和我们的目的基本一致，未登录词识别的准确率较高，而召回率稍低。

微软语料库实验结果数值比较低主要是因为用于训练的《人民日报》语料库切分标准和微软语料库切分标准很多地方存在不一致的情况。在训练集中，人名识

别的姓氏和名字都是切分开来的。但是微软语料库则是合并在一起的。另外，时间、数量词等命名实体的切分标准也有区别。

因为我们的未登录词识别除了分类处理时间和数量词以外，并未按类别分别处理人名、机构名、译名以及新词等这些较复杂的未登录词，所以切分的结果也存在同一类别的词切分不一致的情况。就人名来说，虽然大多数是姓氏和名字切分开的，但是有些人名，如“文忠”、“林格立”因为姓氏容易跟其他字结合成词，所以单字成词的概率较小，导致姓名结合在一起。这种不一致也导致了我们实验结果准确率的下降。实际上有很多判断为错误的切分结果并不影响最终的检索结果。这些词多是因为“词语”这个定义本身的不确定而导致的错误。对这样的词，我们的算法在查询切分和文档切分的结果会保持一致。

4.7 本章小结

本章阐述了针对大数据文本挖掘中的浅层语言分析技术，并对英文浅层语言分析的断句、词汇切分、词性标注以及词形还原进行了阐述，提出了有针对性的改进措施。中文浅层语言分析的特有问题是汉语分词，提出了一种基于层次隐马尔可夫模型的汉语浅层语言分析方法，旨在将汉语分词、词性标注、切分排歧和未登录词识别集成到一个完整的理论框架中。实验表明，层次隐马尔可夫模型的各个层面对汉语词法分析都发挥了积极作用。最后，给出基于单字位置成词概率识别未登录词的算法及其实验结果分析。

相关工作与扩展阅读

词是最小的能够独立活动的有意义的语言成分(朱德熙，1982)。在汉语中，由于词与词之间不存在分隔符，词本身也缺乏明显的形态标记，因此汉语浅层分析的特有问题就是如何将汉语的字串分割为合理的词语序列。从 1983 年第一个实用分词系统 CDWS(梁南元，1987)诞生到现在，国内外的研究者在汉语分词方面进行了广泛研究，提出了很多有效的算法。可以粗略地将这些方法分为两大类：第一类是基于语言学知识的规则方法，如各种形态的最大匹配、最少切分方法，以及综合了最大匹配和最少切分的 N 最短路径方法(张华平等，2002)，还有的研究者引入了错误驱动机制(Hockenmaier et al.，1998)，甚至是深层的句法分析(Wu，1998)。第二类是基于大规模语料库的机器学习方法(Palmer，1997；Dai et al.，1999；高山，2001)，这是目前应用比较广泛、效果较好的解决方案。用到的统计模型有 N 元语言模型、信道-噪声模型、最大期望(Peng et al.，2001)、隐马尔可夫模型等。在实际的分词系统中，往往是规则与统计等多类方法的综合。一方面，规则方法结合使用频率，形成了可训练的规则方法

(Palmer,1997)。另一方面,统计方法往往会采用一些规则排除歧义、识别数词、时间及其他未登录词。同时也注意到国外同行提出的一些探索性算法,如基于压缩的方法(Teahan et al.,2001)、分类器方法(Xue et al.,2002)、无词典的自监督学习方法(Peng et al.,2001)。目前,对不含歧义和未登录词的文本进行分词,已有方法和系统表现相当出色,其准确性已经达到相当高的水平。

实际上,汉语分词的主要瓶颈是切分排歧和未登录词识别。由于切分歧义和未登录词的存在,降低了自身正确切分的可能性,也干扰了其相邻词的处理。如果未登录词和切分歧义交织在一起,就会进一步增加处理难度。例如,在"克林顿对内塔尼亚胡说"中,"内塔尼亚胡"是一个词典中没有收录的译名,实际切分时,"对"与"内","胡"与"说"往往会粘在一起,最终导致错误的切分结果:"克林顿/对内/塔尼亚/胡说/"。在切分歧义方面,文献(Yu et al.,2001)对切分歧义进行了较好的形式化描述,并引入了嘈杂度的概念对歧义的程度进行了定量描述,目前切分排歧的研究路线基本上以规则为主(Kit et al.,2002),还有的针对某一类歧义引入了一些成熟的模型作专门处理,如引入向量空间解决组合歧义问题(Luo et al.,2002)。在未登录词识别方面,主要的出发点是综合利用未登录词内部构成规律及其上下文信息,多采取精细规则处理(Luo et al.,2001),规则一般来源于观察到的语言现象或者是大规模的专名库。目前比较成功的解决方案大都是从大规模的真实语料库中进行机器学习,解决方案有隐马尔可夫模型(张华平,2004;Zhang et al.,2002)、基于 Agent 的方法(Ye et al.,2002)、基于类的三元语言模型(Sun et al.,2002)等。但是,现有的分词方法更大程度上是专门切分出词典中收录的词,基本上没有将未登录词和歧义纳入到同一个算法体系当中,一旦遇到歧义或者未登录词就作为特例进行专门处理,因此使用的模型和方法都没有贯彻到底,缺乏统一的处理算法,对切分结果也缺乏统一的评估体系,最终导致分词的准确率在开放测试的条件下并不像宣称的那样理想,处理含有未登录词、歧义字段的真实文本时,效果更是不尽如人意。这就需要开发一个相对统一的模型框架将排歧、未登录词识别、分词有机融合。这也是本章提出的浅层语言分析的出发点。

近年来,也有不少学者在词法分析方面做了研究。刘秀磊等(2012)针对本体匹配中相似性词法分析算法的不足,提出一种基于词义组合的词法分析算法。该算法首先通过 WordNet 发现本体中单词的合适词义,并扩展它们,然后基于本体的语义元素形式化的定义实体的词法信息标记,最后推理出实体词法信息间的包含关系。针对一组工业本体的测试结果表明该算法有助于提高系统的覆盖率。孟凡东等提出基于统计的中文词法分析往往依赖大规模标注语料,语料的规模和质量直接影响词法分析系统的性能(孟凡东等,2012)。高覆盖率、高质量的语料资源非常有限,而且适用于不同领域的语料往往具有不同的分词和词性标注标准,难以

直接混合使用，从而导致既有资源未能充分利用，分词精度下降等问题。针对该问题，提出了简单有效的异种语料的自动融合方法，并通过实验验证了提案方法的有效性、较强的实用性以及时多种语料融合的可扩展性。孙晓等(2010)针对当前大多数词法分析系统"流水线"式处理方式存在的不足，提出一种一体化同步词法分析机制。在最长次长匹配分词的基础上，在切分有向图中增加词性信息和候选未登录词节点，并拓展隐马尔可夫模型，在切分有向图内同步完成分词、歧义消解、未登录词识别和词性标注等词法分析任务，实现了分词与词性标注的一体化、未登录词识别与分词的一体化以及不确定词性未登录词处理的一体化。一体化机制使词法分析中各步骤实现了真正意义上的同步完成，充分利用上下文词法信息提高整体精度并保证了系统的高效性，避免了各步骤间的冲突。

参 考 文 献

高山，张艳．2001．基于三元统计模型的汉语分词及标注一体化研究．自然语言理解与机器翻译，8：116-122.

梁南元．1987．书面汉语自动分词系统-CDWS．中文信息学报，2：101-106.

刘秀磊，廖建新，朱晓民，等．2012．本体匹配中基于词义组合的词法分析算法．电子学报，40(8)：1624-1630.

孟凡东，徐金安，姜文斌，等．2012．异种语料融合方法：基于统计的中文词法分析应用．中文信息学报，26(2)：3-7，12.

孙晓，黄德根．2010．基于最长次长匹配分词的一体化中文词法分析．大连理工大学学报，50(6)：1028-1034.

张华平，刘群．2002．基于 N-最短路径的中文词语粗分模型．中文信息学报，16(5)：1-7.

张华平，刘群．2004．基于角色标注的中国人名自动识别研究．计算机学报，27(1)：85-91.

朱德熙．1982．语法讲义．北京：商务印书馆.

Brill E. 1992. A simple rule-based part of speech tagger// Proceedings of the Third Annual Conference on Applied Natural Language Processing.

Brill E. 1995. Unsupervised learning of disambiguation rules for part of speech tagging// Proceedings of the 3th Workshop on Very Large Data Base.

Dai Y, Khoo C S G, Loh T E. 1999. A new statistical formula for Chinese text segmentation in corporating contextual information//Proceedings of the ACM SIGIR: 82-89.

Greene B B, Rubin G M. 1971. Automatic grammatical tagging of English. Technical Report.

Hockenmaier J, Brew C. 1998. Error-driven learning of Chinese word segmentation// Proceedings of the 12th Pacific Conference on Language and Information: 218-229.

Kit C Y, Pan H H, Chen H B. 2002. Learning case-based knowledge for disambiguating Chinese word segmentation: a preliminary study// Proceedings of the 1st SIGHAN Workshop, 8: 63-70.

Lawrence R R. 1989. A tutorial on hidden markov models and selected applications in speech rec-

ognition// Proceedings of IEEE, 77(2): 257-286.

Luo X, Sun M S, Benjamin K T. 2002. Covering ambiguity resolution in Chinese word segmentation based on contextual information// Proceedings of the 19th International Conference on COLING: 598-604.

Luo Z, Song R. 2001. Integrated and fast recognition of proper noun in modern Chinese word segmentation// Proceedings of International Conference on Chinese Computing: 323-328.

Manning C, Hinrich S. 1998. Foundations of statistical natural language processing. New York: MIT Press.

Palmer D. 1997. A trainable rule-based algorithm for word segmentation// Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics.

Peng F, Schuurmans D. 2001. A hierarchical EM approach to word segmentation// Proceedings of the 6th Natural Language Processing Pacific Rim Symposium.

Shai F, Yoram S, Naftali T. 1998. The hierarchical hidden markov model: analysis and applications. Machine Learning: 32-41.

Sun J, Gao J F, Zhang L, et al. 2002. Chinese named entity identification using class-based language model// Proceedings of the 19th International Conference on Computational Linguistics.

Teahan W J, Wen Y, McNab R, et al. 2001. A compression-based algorithm for Chinese word segmentation. Computational Linguistics, 26(3): 375-393.

Viterbi A J. 1967. Error bounds for convolutional codes and asymptotically optimal decoding algorithm. IEEE Transactions on Information Theory, 13: 260-269.

Wu A D, Jiang Z X. 1998. Word segmentation in sentence analysis// Proceedings of the International Conference on Chinese Information Processing.

Xue N W, Susan P C. 2002. Combining classifiers for Chinese word segmentation// Proceedings of the 1st SIGHAN Workshop.

Ye S R, Chua T, Liu J M. 2002. An agent-based approach to Chinese named entity recognition// Proceedings of the 19th International Conference on Computational Linguistics.

Yu J S, Yu S W. 2001. Some problems of Chinese segmentation// Proceedings of the 1st International Workshop on Multi Media Annotation.

Zhang M, Song R, Lin C, et al. 2002. Expansion-based technologies in finding relevant and new information: THU TREC2002 novelty track experiments// Proceeding of the 11th Text Retrieval Conference.

第 5 章　大数据语言新特征发现

5.1　概　　述

信息的基本单位是句子，一个句子可以较完整的表达连贯的易于理解的语义。句子中起主要作用的往往是关键词、词组或短语，而句子中的其他成分只是起到进一步修饰连接的作用，它们只是对基本信息的细化和补充。一旦获得这些有意义的关键信息，便能获取句子的基本信息。因此，新特征语言和有意义串发现对中文自然语言理解是很有意义的。发现频繁使用的新词和短语等有意义串，如“黑砖窑事件”、“次贷”、“大小非”、“D 字头”、“奥运钞”、“山寨”、“打酱油”、“叉腰肌”、“艳照门”、“凤凰男”、“俯卧撑”、“自主创新”、“十二五规划”、“和谐社会”、“社会主义新农村”、“社会主义荣辱观”等，不仅能改进分词准确率，也有广泛的应用前景。其应用领域主要如下几个方面。

第一，是对信息检索索引词的分析。例如，“计算技术研究所”是一个完整的查询词，而“计算”或者“技术”几乎不会用来去查询。当用户想搜索大众汽车时，输入“大众”得到的搜索结果可能不太准确，包含很多类似“人民大众”、“劳苦大众”这样非大众汽车的信息，而作为有意义串的“上海大众”具有语义完整性，能够排除歧义，一般能更准确地描述用户的需求。所以，有意义串能够应用到信息检索的查询词修正以及相关的搜索分析。

第二，在社会热点挖掘与跟踪方面有应用价值。一个有意义串就是一个线索，就是一个很有价值的社会现象。它们往往包含了网民对当前社会各种现象的立场和观点。因此，对新特征语言和有意义串的挖掘，对社会热点挖掘与舆情监控的意义重大。

第三，能用于信息分析与特征提取。除字和词外，常用的文本特征项主要有短语、语义概念、有意义串等。有意义串的提取对于改善文本分类和聚类性能具有重要指导意义。

第四，对词典扩充与语料库建设意义重大。由于新词出现的速度不断加快，新词出现的领域不断扩大，使用传统的人工方法搜集新词语，既耗时又耗力且时效性差。如果利用电脑的计算能力和自动检测方法，可以快速输出新词候选供人工进行筛选，这将大大减轻人的负担。若能将新词属于有意义串的一部分自动提取出来，必将促进语料库的自动化建设。此外，有意义串挖掘进一步深化可以进行关键频繁模式挖掘，对更高层次的文本自动内容抽取、话题检测和机器翻译等应用都有

比较重要的意义。

新特征词和有意义串是指具备统计意义的串，新特征词和有意义串主要分为以下几个类别(其中前两类全部是词语，后三类中既包括词语又包括短语和词组)。

(1) 命名实体，如“陈水扁”、“民进党”、“巴西队”等。

(2) 新造词，指狭义新词，如“博客”、“户型”、“低保”、“撞衫”等。

(3) 领域术语，指领域相关的常用术语，如“计算语言学”、“未登录词”、“股骨头坏死”等。

(4) 固定搭配，主要指通用语料中经常使用的常用搭配，如“台湾同胞”，“住房需求”等。

(5) 成语、歇后语等惯用语，如“智者千虑，必有一失”，“仁者见仁，智者见智”等。

对新特征语言和有意义串挖掘是不容易的，主要表现在以下方面。

(1) 命名实体以及新词等未登录词属于开放集合，在实际生活中不断产生，数量巨大，无法枚举。

(2) 命名实体种类繁多，包括数词、人名、地名、机构名、译名、时间、货币等。有些词，如时间、货币等具有很强的规律性，较容易识别，而另外一些词没有明显规律，识别困难，且常以缩写形式出现，没有简单的规则能够覆盖众多类型的命名实体和新词。

(3) 网络上经常出现一些不规范用语，且未登录词和常用词混在一起，分词时也容易引起歧义，如“[王辉[国]家]里有点急事”、“发烧[是[非]典]的典型症状之一”，这些都给识别带来了困难。

(4) 在短文本(如微博)中进行联机识别尤其困难，因为在短文本中，未登录词不会频繁出现，简单的统计方法很难识别出这些低频词语。

现已有很多学者用统计的方法提取有意义串，即根据一个字符串的频率、互信息、邻接类别等统计量来判断该串是否为有意义串。这种方法对高频与意义串处理结果较好，但往往很难有效地提取低频的有意义串。有的学者发现意义串倾向于在语料局部集中出现，该特性有利于提取语料中的低频串。如果某个字符串在语料的局部集中出现，即使其频率等属性相对较小，也可以认为该字符串是有意义串。在这个局部区域，该字符串具有相对较高的频率等属性，因此有意义串的这种局部性特征能够提取其他统计量很难有效提取的低频串。

本章从上下文邻接分析、双字耦合度、单字位置成词概率、基于局部性分析与伪相关反馈模型的有意义串挖掘算法等几个不同角度出发，采用不同策略组合进行实验并给出评价结果。设计了一系列不同的实验对比分析提取过程中各因素对结果的影响，同时验证了算法的有效性。算法流程如图 5.1 所示。

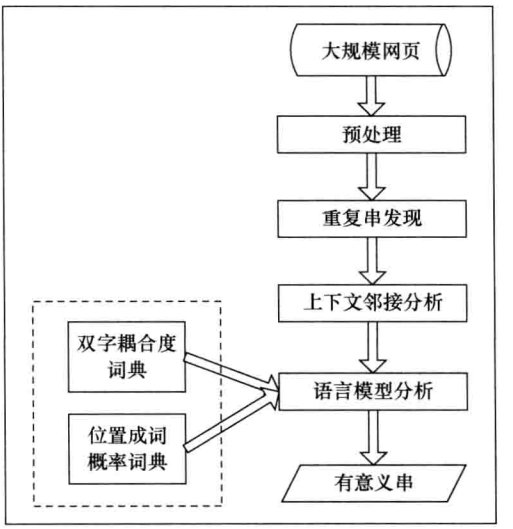


图 5.1 算法流程

5.2 基于上下文邻接分析和语言模型的有意义串提取

已知词已经作为频繁使用的语言单位固定下来，在词典中收录，不需专门提取，因此对有意义串和新特征语言的提取工作重点就是发现核心词典中尚未收录的广泛使用的有意义词和短语等。

在语料处理阶段已经得到格式化的纯文本文件并完成了分词、将汉字转化为对应的 ID 等工作。大规模语料中统计出来的重复串中除了包含有意义串外，还存在大量的垃圾串。新特征词和有意义串提取的实质就是在重复串发现的基础上过滤掉无意义的垃圾串。垃圾串一般分为三类：第一类是冗余子串，像“邓小平理”这样的串完全包含在“邓小平理论”中；第二类是有意义串和常用字的组合，如“在批发市场”、“胡锦涛说”；第三类是频繁功能串，如“这是”、“我的”等。对第一类垃圾串的判断需要依赖它的外部使用环境，采用上下文邻接分析识别。上下文分析是从串的上下文入手分析其使用环境，主要通过上下文邻接种类、邻接熵、邻接对种类以及邻接对熵等邻接特征量来判断。后两类垃圾串的判断要从串本身结构着手，通过语言模型（主要采用双字耦合度及字的位置成词概率）来判断、分析这些串的语义是否完整。本节将在解释邻接种类、邻接熵、邻接对种类、邻接对熵、双字耦合度、位置成词概率等概念及性质之后，给出新特征语言和有意义串提取的完整流程。

5.2.1　上下文邻接分析

作为频繁使用的语言单元，通用的已知词语在实际生活中具有一定的流通度和分布广泛性。通常来说，一个词语的内部结合度比较高，如“尴尬”、“匍匐”中的两个字总是一起出现，它与外部上下文的关系比较松散。有意义串作为独立使用的语言单元，在真实文本中也有一定的流通度，能够应用于多种不同的上下文环境，而非某种特殊语境下的临时性组合。为了描述字符串 S 的上下文语境灵活程度，定义一系列邻接特征量。

定义 5.1 邻接集合 NB　分为左邻接集合 NBL 和右邻接集合 NBR，分别指真实文本中，与字符串 S 左边或者右边相邻的元素集合。元素可以是字、词等不同粒度的语言单位，当字符串作为一个句子的开始时，其左邻接元素记为 BOS；作为句子的结束时，其右邻接元素记为 EOS。

定义 5.2 邻接种类 V　分为左邻接种类 VL 和右邻接种类 VR，分别指左邻接集合中和右邻接集合中元素的数目，它们反映了串 S 上文和下文语境种类的多少。

邻接种类在一定程度上反映了字符串的语言环境，只有当一个串的左邻接种类和右邻接种类都满足一定条件时，该串才具有语用灵活性。如果某个邻接种类值为 1，另一个较大，则通常情况下该串为某个串的字串。但邻接种类忽略了邻接集合中每个元素出现的频次，它只记录了出现与不出现两种情况。在实际文本中，每个邻接元素的出现频次不同，其分布规律与语用灵活性有一定的关系。

信息论中的信息熵是信息度量的基本单位，表征了信源整体的统计特征，是总体平均不确定性的度量。我们把邻接集合的信息熵称为邻接熵。

定义 5.3 邻接熵　表示字符串 S 的邻接集合的信息熵，邻接集合 NBL 中每个元素 l_i 在真实文本中对应一个出现频次 n_i，频次总和记为 N，邻接熵的计算公式为

$$E_L = -\sum_{i=1}^{|V_L|} \frac{n_i}{n} \log\left(\frac{n_i}{n}\right)$$

熵值反映了信息的不确定性，而邻接熵则反映了字符串 S 邻接元素的不确定性。熵值越大，表明邻接元素的不确定性越大，字符串 S 的语用环境越灵活，从而它是有意义串的可能性也越大。邻接熵结合了字符串上下文的语境种类及其频次，比邻接种类包含更多的信息量，判断语用灵活性时它是一个比邻接种类更强的特征量。当语料规模较小时，左右邻接种类本来就比较少，计算得到的熵值区别力不大。

前面的定义是对字符串的上文和下文独立并分析，分别定义了左邻接和右邻接的特征量。左邻接和右邻接是一个串完整的上下文环境，可以结合起来作为一

个整体来研究。下面给出邻接对集合，邻接对种类以及邻接对熵的定义。

定义 5.4 邻接对集合 字符串 S 每次出现的左邻接元素和右邻接元素构成一个邻接对$<L_i,R_i>$，字符串 S 的所有邻接对组成邻接对集合 PNB。

定义 5.5 邻接对种类 邻接对集合 PNB 中元素的个数称为邻接对种类 VP。

定义 5.6 邻接对熵 表示邻接对集合的信息熵，邻接对集合 PNB 中每个邻接对$<L_i,R_i>$在真实文本中对应的频次为 n_i。

一个串的邻接对集合显示了该串在文本中完整的语用环境，对于研究有意义串的词性以及其他性质具有重要作用。例如，新词“禽流感”从 2004 年开始频繁使用，下面是新闻网页中的几个真实例子。

(1) 钟南山**透露**禽流感**病毒**尚未明显变异。

(2) 广东的**防控**禽流感**形势**趋缓。

(3) 发生了 7 起人**感染**禽流感**事件**。

(4) 发现**一宗**禽流感**疑似**病例。

(5) 颁布 5 条禁令防控禽流感。

如果将词作为邻接分析的粒度，“禽流感”这个串的邻接特征量计算结果如下。

(1) 左邻接集合 NBL={**透露**，**防控**，**感染**，**一宗**}(加粗带下划线显示)。

(2) 右邻接集合 NBR={**病毒**，**形势**，**事件**，**疑似**，**EOS**}(加粗带下划线显示)。

(3) 左邻接类别 VL=4。

(4) 右邻接类别 VR=5。

左邻接熵

$$E_L=\frac{1}{5}\log\frac{1}{5}+\frac{2}{5}\log\frac{2}{5}+\frac{1}{5}\log\frac{1}{5}+\frac{1}{5}\log\frac{1}{5}=0.718$$

右邻接熵

$$E_R=\frac{1}{5}\log\frac{1}{5}+\frac{1}{5}\log\frac{1}{5}+\frac{1}{5}\log\frac{1}{5}+\frac{1}{5}\log\frac{1}{5}+\frac{1}{5}\log\frac{1}{5}=0.699$$

邻接对集合

PNB={<透露，病毒>，<防控，形势>，<感染，事件>，<一宗，疑似>，<防控，EOS>}

邻接对熵

$$E_P=\frac{1}{5}\log\frac{1}{5}+\frac{1}{5}\log\frac{1}{5}+\frac{1}{5}\log\frac{1}{5}+\frac{1}{5}\log\frac{1}{5}+\frac{1}{5}\log\frac{1}{5}=0.699$$

可以看出，“禽流感”这个串的语言环境变化多样，已经构成一个独立应用，是不依赖于上下文的语言单元。我们在统计中也发现“珠三”(即珠江三角洲)这样的串，左邻接类别高达 9，而右邻接类别仅为 1，只有上文环境比较灵活，而实际上该串的出现完全依赖于另一个串“珠三角”，所以只有上文或者下文语用显示出多样

性的串往往是冗余字串。所以，一个串能够成为有意义串除了要满足统计特性，还要结构稳定，上文和下文环境同时具有一定的多样性。

5.2.2　语言模型分析

1. 双字耦合度

汉语中常用的汉字有六千多，常用词(包括双字词和多字词)有八万多，这说明汉语语言有一定的构词规律与用语习惯，并非任何几个字或多个字组合在一起都可以构成词。例如，“王二小”是作为一个人名很普遍，但“王二大”这样的名字很少出现。为了描述一个词中连续两个字的结合紧密程度，引入双字耦合的概念。

定义 5.7 双字耦合度　在经过分词切分好的训练语料中扫描所有出现过的连续双字对，统计出每组字对出现的总次数 N_1 以及该字对作为某个词子串的总次数 N_2，后者与前者之比就叫做双字耦合度 Coup，计算公式为

$$\mathrm{Coup}(<c_i,c_{i+1}>)=\frac{N(^{*}c_m\cdots c_ic_{i+1}\cdots c_n^{*})}{N(^{*}c_m\cdots c_ic_{i+1}\cdots c_n^{*})+N(^{*}c_m\cdots c_i^{*}\ c_{i+1}\cdots c_n^{*})}$$

其中，$*$ 表示切分标记；$N(^{*}c_m\cdots c_ic_{i+1}\cdots c_n^{*})$ 表示 c_ic_{i+1} 作为某个词子串出现的频次；$N(*c_m\cdots c_i^{*}c_{i+1}\cdots c_n^{*})$ 表示 c_ic_{i+1} 相邻的两个字分别属于两个词的频次。

例如，“过目”这一双字对在文的统计中共出现 16 次，其中出现“过目不忘”，“一一过目”这样的词中 12 次，而在“超过/目前”这样的语境中共出现了 4 次，所以 Coup(<过，目>)=12/(12+4)=0.75。表 5.1 是双字耦合度示例。

表 5.1　双字耦合度示例

字对	N_1	N_2	Coup($<c_i,c_j>$)
好看	52	50	0.9615
德昂	8	8	1
否具	10	0	0
才基	30	1	0.0323
化气	31	18	0.5806
所思	18	8	0.4444

双字耦合度这一特性反映了字与字之间的内部构词特征，是字与字之间构词能力的体现，具有如下性质。

(1) 任何字对的 Coup 值介于[0,1]之间，越接近 1，表明该字对的结合程度越高，越接近 0，表明该字对越不可能出现在一个词中。

(2) 字对是有序的，$<c_i,c_{i+1}>$的耦合度与$<c_{i+1},c_i>$的耦合度是完全独立的。

(3) 耦合度分布极其不均匀。通过对《人民日报》1998 年 1～5 月的已标注语料(共 48.7M)做了统计，得到 605 918 组不同的字对。但发现有大量的字对(30

多万)出现频次小于 3,只能视作小概率事件,经过平滑处理应该将其去掉。最后得到的统计结果只剩下不到 25 万的频繁出现字对,具体概率分布如表 5.2 所示,其中 Range 表示双字耦合度的范围,Num 是处于该范围内的字对数目,Percent 表示该范围的字对数占字对总数的百分比。

表 5.2 双字耦合度分段分布表

Range	[0,0.1)	[0.1,0.2)	[0.2,0.3)	[0.3,0.4)	[0.4,0.5)
Num	171 817	3684	2591	3827	2440
Percent/%	**71.38**	1.53	1.07	1.59	1.01
Range	[0.5,0.6)	[0.6,0.7)	[0.7,0.8)	[0.8,0.9)	[0.9,1]
Num	1594	3365	3571	3703	44 466
Percent/%	0.66	1.40	1.48	1.54	**18.56**

从表中可看出耦合度的分布状况不均匀,小于 0.1 的字对超过 70%,大于 0.9 的也占到 18.56%,只有大约 10%的少数字对耦合度介于二者之间,呈现出两头多,中间少的特点。这种分布不均匀特性决定了在处理各分布区域的字对时不能采用统一的方法来做判断。这里采用了双字耦合度分段策略,给定阈值 C_{max} 和 C_{min},对于已知的字对,按如下规则判断其是否构成有意义串的一部分:

(1) 若耦合度大于阈值 C_{max},则说明它们构成某个词的字串,不是垃圾串组合,应该保留。例如,表 5.1 中的"好看"和"德昂"这两组字对通常出现在一个词中。

(2) 若概率小于某个阈值 C_{min},则认为它们通常不出现在一个词中,应该切分成不同的词,是垃圾串。例如,"否具"、"才基"这样的组合往往是要分开的。

(3) 若概率介于 C_{max} 和 C_{min} 之间,反映了该字对结合和分开的情况相当,目前学习到的内容暂不能成为判断是否是垃圾串的依据,不做分析。例如,表 5.1 中的"化气"和"所思"。

2. 单字位置成词概率

汉语中的每个字都有自己特有的构词用法,并非所有的字都可以作为词首、词中、词尾的每种成分,某些字往往出现在合成词的某个或某几个特定位置上,像"老虎"、"老师"中的"老"通常出现在词首,"先进性"、"优越性"中的"性"通常出现在词尾,而另为一些字总是不会出现在词首或者词尾,如"尴"、"蚓"。文献(Wu,2000;王小飞,2006)用位置成词概率来描述之。这里主要关注词首/词尾成词概率。

定义 5.8 位置成词概率 在切分好的语料中统计每个汉字出现在词语中的总次数 N 和出现在词首、词尾的次数 N_1 和 N_2,那么词首成词概率是 N_1 与 N 的比值,词尾成词概率是 N_2 与 N 的比值。计算公式(pos 取值范围为 0 和 1,分别表示词首和词尾)为

$$P_{\text{word}}(c,\text{pos})=\frac{N(c,\text{pos})}{N(c,\text{word})}$$

这里将《人民日报》1998 年 1～5 月的已标注语料(共 48.7MB)作为训练语料，得到 6700 多个常用汉字的词首以及词尾成词概率。部分统计如表 5.3 所示。

表 5.3　位置成词概率示例

汉字	词首概率	词尾概率
阿	0.875879	0.009172
士	0.04307	0.81512
田	0.550494	0.451128
径	0.026696	0.915462
老	0.75296	0.13562
性	0.062986	0.923597

有意义串的开始是某个词的词首，串尾也一定是某个词的词尾。当检测到串首某个字的词首成词概率太低时，该串可能是垃圾串；若串尾某个字的词尾成词概率太低时，该串也可能是垃圾串。此条规则能有效地过滤新词和常用字的组成这一类垃圾串，如判断“胡锦涛说”中的“说”，“在批发市场”的“在”，判断它们的位置成词概率即可将垃圾串排除。此外，还有一部分频繁功能串也能被过滤掉，如“我的”、“这是”等高频串。

5.2.3　重复串发现及处理流程

重复串发现是对 n 元递增算法作了改进，主要分为以下几个步骤。

(1) 对处理好的 ID 序列建立索引。对于只有 5 个字符的文本“abcabdabaeaabc”，对该文本建立单字索引如图 5.2 所示。

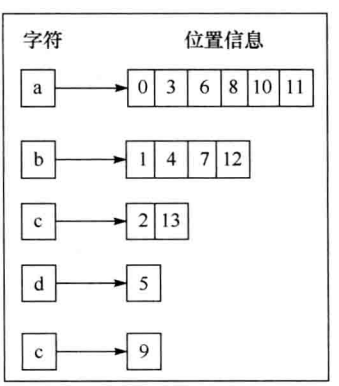


图 5.2　索引示例

(2) 从每个单字索引的信息开始扩展得到所有重复串。因为单字索引过程中已记录好语料中所有出现 a 的地址集合 A，而以 a 打头的所有重复串的地址一定包含在集合 A 中，那么遍历地址集合 A，根据地址扫描语料进行二字扩展、三字扩展……，可以发现以 a 打头的所有重复串。在上面的例子中，扫描所有 a 后面的第一个字符，扩展得到以 a 打头的所有两字串(图 5.3)和三字串(图 5.4)。

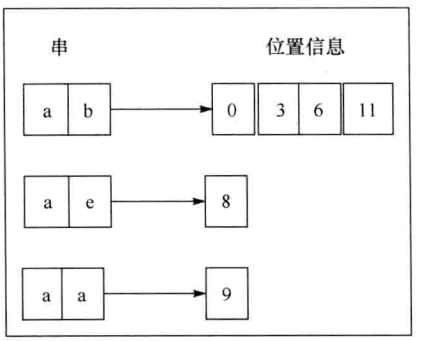


图 5.3 二字扩展示例

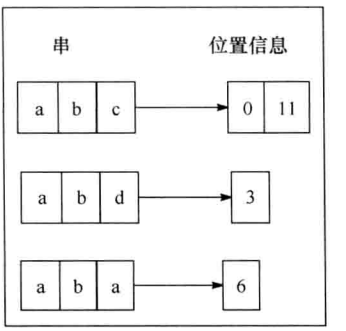


图 5.4 三字扩展示例

频次为 1 的串(如 ae、aa)不是重复串，将其删除；对于频次大于 1 的重复串(如 ab)，要将所有信息写入重复串文件中且继续对其扩展，发现长度增 1 的重复串。

(3) 新产生的重复串写入文件后继续扩展得到长串，反复迭代，直到出现间隔符号或长度达到指定阈值。

在有意义串分析中需要利用串的文档信息和邻接对信息。如果发现重复串后再进行上述统计，则需要对整个语料做多次遍历，势必增加时间开销。如果在发现重复串之外，记录每个串的邻接词信息以及文档信息并独立保存在一个文件中，将提高处理效率。首先，开辟一个缓冲栈用于存放当前扩展串及相关信息，开始时读入单字索引，根据其地址链查找原始 ID 序列进行扩展，扩展串中频次大于 1 的串被认为是重复串，统计其文档和邻接对信息后输出到文件中，同时放入缓冲栈中用于以日后展。当一次扩展结束后再从栈中读取栈顶串进行新一轮扩展。第一次栈空时表示以第一个 ID 打头的所有重复串发现完毕，然后读取第二个字的索引链并放入缓冲栈，依次反复直到所有重复串发现完毕。最后输出一组文件，重复串文件用来存储串的内容、串长、串频、文档频次、邻接对频次等信息；地址链文件、文档链文件、邻接对文件则分别按序存放与重复串文件中各串相对应的地址、文档信息、邻接对信息等。总体流程如图 5.5 所示。

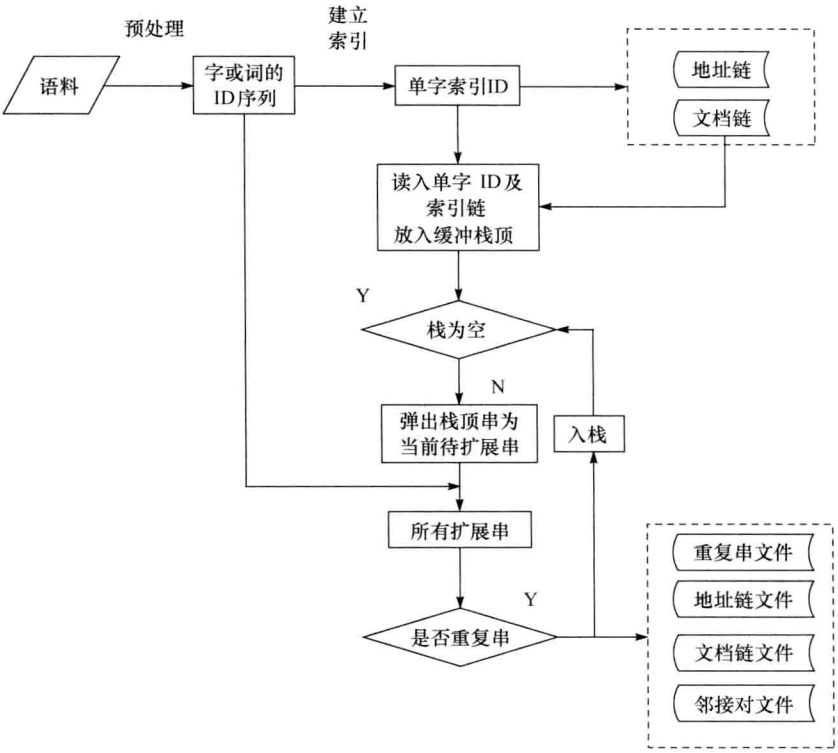


图 5.5　重复串发现流程

如果一个串是有意义串，需要它在结构、语用、语义以及统计方面满足一定的特点。但上述重复串发现产生的结果只满足了统计性要求，可称为候选有意义串，而它们是否具有有意义串的其他性质还要进一步判断。接下来将通过上下文邻接计算来分析串的语用情况，通过语言模型来判断其在语义上是否完整。详细流程如图 5.6 所示。其中的“上下文邻接分析”主要是计算每条重复串的上下文邻接特征量，包括邻接类别、邻接对类别、邻接熵、邻接对熵等，判断这些特征量是否达到设定的阈值，若未达到阈值，则说明该串是垃圾串，将其过滤掉；如达到，则说明该串在语用上比较灵活，进入下面的语言模型分析。

语言模型主要分析从串的内部结构来考察其是否能构成完整语义。在语言模型分析之前要对一部分训练语料进行人工标注，根据公式计算生成双字耦合度词典和单字位置成词概率词典。实验中的双字耦合度和单字位置成词概率是在《人民日报》1998 年 1～5 月语料上训练得到的。语言模型分析时，首先逐字扫描相邻两字对，查找其双字耦合度，小于设定阈值 $C_{\min}$ 时（如“否具”、“才基”等）不构成某词的一部分，作为垃圾串删除。之后，查找单字位置成词概率，来判断其串首和串尾是否包含常用功能字。流程如图 5.7 所示。

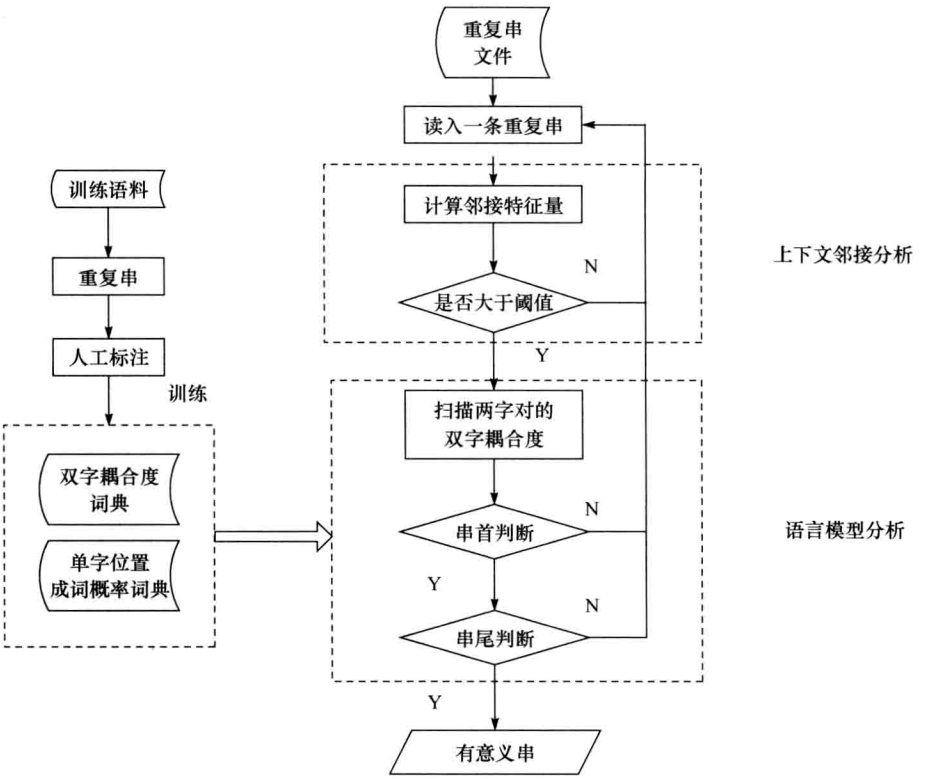


图 5.6　重复串中提取有意义串流程

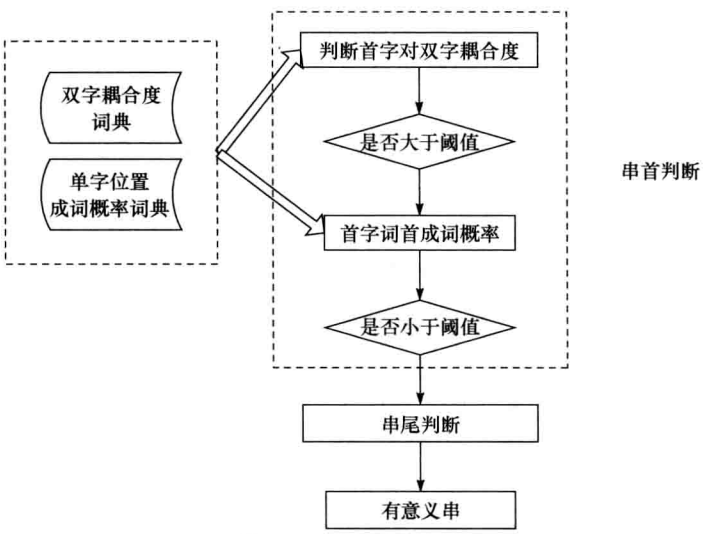


图 5.7　串首串尾判断

取出串中首字对，判断其双字耦合度，如果大于阈值 C_{max}，则该字对结合紧密，构成某个词的首部，不再对首字的单字位置成词概率进行判断，这样能避免垃圾头词典的绝对化问题。例如，“的士”这个双字对是一个词，仅仅判断首字“的”的位置成词概率，也许需要过滤，但是首先判断字对的双字耦合度，发现其耦合程度高，应保留。只有首部双字对的耦合度没达到指定阈值 C_{max} 时才判断首字位置成词概率，如概率小于 P_{begin}，说明首字不适合作为一个词的首部，也要将其过滤。通过词首判断而保留的串继续判断词尾是否包含常用功能字，其过程类似词首判断，要将双字耦合度和词尾成词概率结合起来分析。在词首和词尾判断中没有被删除的串最后将作为有意义串输出到文件中。

5.2.4　实验设计及结果分析

实验语料采用 News-Corpus(470MB)语料，在《人民日报》1998 年 1～5 月的已标注语料(48.7MB)训练得到双字耦合度词典和单字位置成词概率词典。实验一方面验证上下文邻接分析与语言模型分析方法的有效性及可行性，另一方面测试该方法处理大规模语料的能力。下面的前五个实验语料取自 News-Courpus 语料中的 4783 个文档，最后一个实验采用 470M 的 News-Coupus 全部语料。在实验中各参数的取值如表 5.4 所示。

表 5.4　实验参数取值

参数名	作用	取值
C_{max}	双字对结合阈值	0.95
C_{min}	双字对分隔阈值	0.30
P_{begin}	单字词首判断	0.45
P_{end}	单字词尾判断	0.50

1. 对比不同策略在识别过程中所起的作用

上下文邻接分析、双字耦合度、单字位置成词概率是识别过程中的三个不同策略。我们分别采用不同策略的组合来做实验，预处理阶段不做分词，表 5.5 列出了所有策略单独使用以及所有策略组合产生的实验结果实验。RF 指重复串发现(repeat find)，CA 指上下文邻接分析(context analysis)、CD 指采用双字耦合度策略(coupling degree)、IWP 指采用单字位置成词概率(inside word probability)。

表 5.5 不同策略组合实验结果

实验序号	实验条件	结果数目	正确结果	准确率/%	召回率/%	值/%
1	RF	76 399	2350	3.08	100	5.97
2	RF+CA	17 784	1750	9.84	74	17.38
3	RF+CD	65 625	2309	3.52	98	6.79
4	RF+IWP	40 607	2158	5.31	92	10.05
5	RF+CA+CD	11 773	1728	14.68	74	24.47
6	RF+CA+IWP	7802	1636	20.97	70	32.23
7	RF+CD+IWP	36 492	2134	5.85	91	10.99
8	RF+CA+CD+IWP	3640	1673	45.96	71	55.86

图 5.8 显示出了三种策略单独使用时的结果对比，重复串(RF)是实验结果对照的基准，从 F 值来看，上下文邻接分析是最有效的，因其过滤大量垃圾串的同时只损失了一小部分正确结果；双字耦合度方法召回率最高，因为训练学习到的双字耦合度非常有限，识别时只对已学习到的信息做出判断，而未学习过的双字继续保留；单字位置成词概率比双字耦合度更有效，因为有意义串与常用字组合这样的垃圾串是重复串中数量最多的垃圾串。

图 5.9 显示三种策略两两组合时的实验结果，可以看出包含上下文邻接分析的两种方法准确率和 F 值都比未使用上下文邻接分析的方法要高。这说明上下文邻接分析先于语言模型分析是合理的，能够在排除大量垃圾信息，而语言模型分析是对上下文邻接分析漏掉的垃圾串做补充过滤。

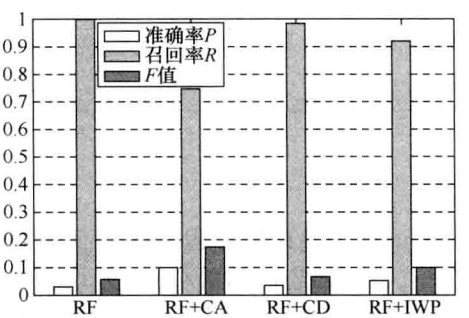


图 5.8 单独策略结果图

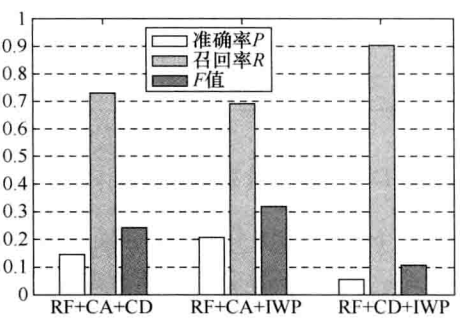


图 5.9 双策略结果图

图 5.10 是三种策略逐步添加的实验结果图。可以看出，每步过滤操作都在召回率下降不多的情况下准确率得到大幅度提高。说明每种过滤策略都发挥作用。经过上下文分析后，漏掉一部分正确有意义串，这是因为实验语料规模较小，有些有意义串仅仅出现两次，而两次出现的上下文环境又相同，邻接特征量太低，不具

有语用灵活性而丢失。在大规模语料上实验的重复串频次的设置可以增大，能够削弱此问题。

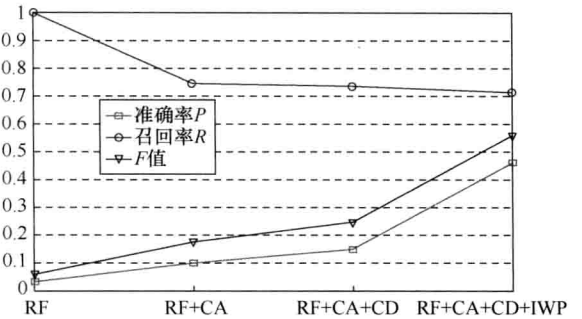


图 5.10　逐步添加策略实验结果图

2. 对比邻接分析粒度对实验结果的影响

上下文邻接分析的粒度可以是字和词。本实验是在预处理阶段分词(分词采用速度较快的最大匹配分词法，分词词典只包括了 6 万核心词语)与不分词的情况下，分别对比邻接分析粒度的大小对实验结果的影响。实验结果如表 5.6 所示。

表 5.6　邻接粒度结果对比

上下文邻接分析粒度	结果数目	正确结果	准确率 P/%	召回率 R/%	F 值/%
字(未分词)	3640	1673	45.96	71.19	55.86
词(分词后)	2533	1787	76.55	79.04	77.78

从表中看出，分词后的性能在正确率和召回率方面比没有分词的有明显改善，主要原因是分词后避免了串中所谓的“半个无意义词”的出现，如“社会主”出现在“社会主义”、“社会主人”和“社会主张”等词中。它在实验一中可能被识别为新词，而由于“主义”、“主人”和“主张”等都是分词词典中的已知词，因此不会有所谓的“半个无意义词”的现象出现。

3. 上下文邻接分析阈值变化对实验结果的影响

在进行上下文邻接的性能分析时，特征量的阈值选取会对实验结果有影响，这里给出邻接种类阈值 T_{CA} 从 1～8 变化时的实验结果。从图 5.11 可以看出，随着 T_{CA} 的增大，正确率逐步增长，召回率在逐步下降。当阈值达到 4 时，正确率已超过 90%。F 值一直在下降，可能是因为本实验的实验规模较小，频度是 2 的重复串几乎占到所有重复串的一半，T_{CA} 增大将漏掉大量的正确结果，致使召回率的下降比正确率增长的影响程度大。当实验语料规模增大时，随着阈值的增大召回率

下降将不至于太快，F 值曲线将会先上升再下降，且在针对网络大数据的研究中，准确率比召回率更加现实。

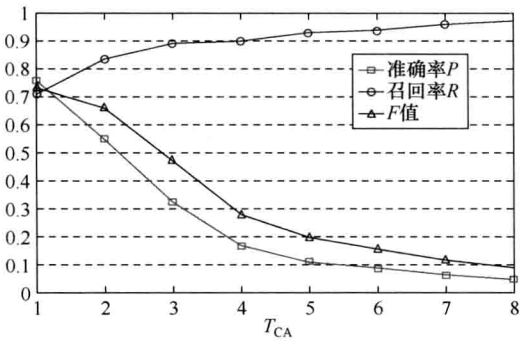


图 5.11 邻接种类阈值变化对实验结果的影响

4. 对于不同长度的有意义串的实验结果对比

有意义串的长度分布是不均匀的。长度是 4 以下的串占到总数的 70%以上，长度为 4 的数量最多，占 40%以上。本实验统计不同长度有意义串的正确率和召回率，分析串长度与实验结果的关系，实验结果如图 5.12 所示。可以看出，长度为 2 的正确率最低，但召回率最高，这可能是因为长度为 2 的重复串数量最多且同时包含的常用功能串也最多，而过滤时，不同长度字串的参数设置是一致的，对于长度为 2 的串过滤规则不足够强所致。从整条曲线来看，正确率处于上升趋势，召回率处于下降趋势，因为随着长度的增长，重复串数量递减，而垃圾串的比例也在递减。

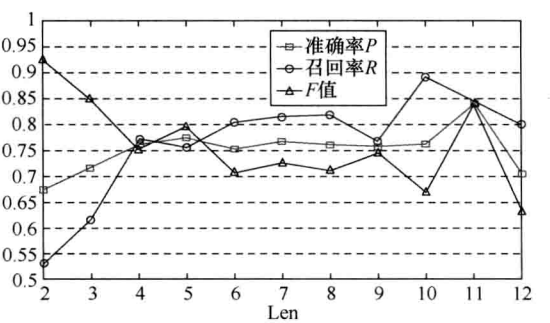


图 5.12 不同串长实验结果

5. 不同邻接特征量对实验结果的影响

在上下文邻接分析中，可采用邻接类别、邻接熵、邻接对类别、邻接对熵等不同的邻接特征量。实验中已经观察到邻接特征量的阈值设置不同，对实验结果的影响较大，而由于邻接类别和邻接熵的量纲不同，无法设定统一的阈值来衡量实验结果，所以这里通过 $P@N$ 指标来对比结果优劣，如图 5.13 所示。

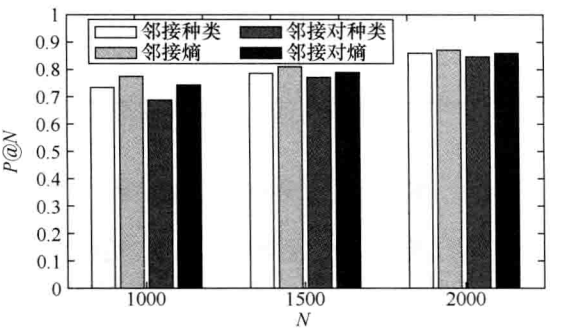


图 5.13　不同邻接特征量结果对比图

图中显示了 N 取 2000、1500、1000 时的三组结果，每组包括采用邻接种类、邻接熵、邻接对种类、邻接对熵，来分析串的语用灵活性的实验结果。从每一组结果内部可看出，邻接熵的效果一般优于邻接种类的效果，邻接对熵的效果优于邻接对种类，所以衡量语用环境的多样性方面，熵值比种类有效些；邻接类别比邻接对类别结果好；邻接熵结果比邻接对熵结果好，说明将左邻接和右邻接独立开来分析比作为邻接对分析时的效果好，因为邻接对于一侧语用灵活而另一侧语用固定的冗余子串无法过滤，且邻接对较稀疏。

6. 大数据语料上的实验

在前面的小数据语料实验中，已经初步显示出邻接分析与语言模型分析相结合的有意义串提取算法的有效性。下面将在 New-Corpus 的 470M 全部语料上进行实验，来分析其在大数据语料上的提取性能。本实验结果多达几十万条，而人工统计正确率和召回率不太现实，所以对实验结果按照一定策略排序后，计算 $P@N$ 正确率，排序的依据有频次和邻接类别两种，结果如图 5.14 所示。

图中显示了 N 从 200 增大到 2000 时，分别按照频次 F 和邻接类别 VNB 两种方式排序时 $P@N$ 的变化情况。通过对两条曲线的对比分析发现，按照邻接类别排序比按照频次排序更有效，同时也反映了语用多样性是有意义串的固有属性。从结果来看，N 从 200 变化到 2000 的过程中，结果准确率一直在 95%附近，基本接近实用化的水平。只从某条曲线来分析，呈现出了先上升后下降的趋势，随着

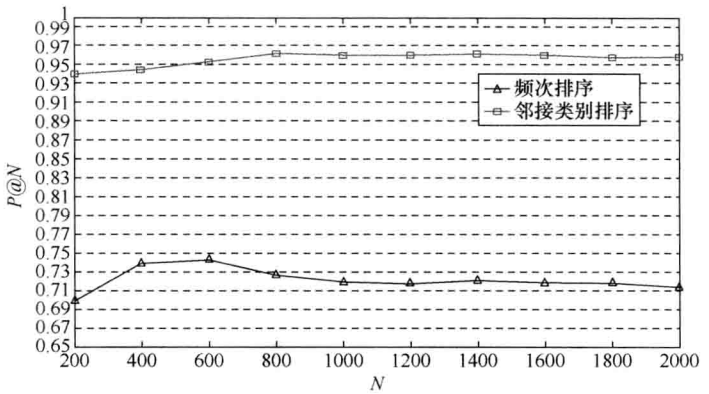


图 5.14 大规模语料实验结果

N 的增大，$P@N$ 逐渐降低，原因是因为排序越靠后，是有意义串的可能性越小，正确率自然越低，而曲线一开始出现一点上升是因为常用的部分功能垃圾串是最高频的，它们没有全部过滤掉，而出现在高频结果中，使得 N 值较小时正确率反而低。

前面几个实验分别验证了本章提出的算法在大小规模不同的语料上都是有效的，小规模实验的准确率为 76.55%，大规模实验的 $P@N$（N 为 2000）结果超过 95%。此外，还从实验中发现了实验过程中不同因素对实验结果的影响。

（1）上下文邻接分析反映了串的语用灵活性，能够有效过滤垃圾串，在语言模型分析之前进行是合理的。

（2）上下文邻接、双字耦合度和单字位置成词概率每一种过滤策略都起作用，不可或缺。

（3）词语作为邻接分析的粒度比字更有效。

（4）邻接熵作为邻接特征量比邻接邻接种类效果更优。

（5）一个串的邻接种类越大，表示语用环境越灵活，越可能是有意义串。

5.3 基于局部性原理的低频有意义串提取

虽然可以用统计的方法提取有意义串，如根据一个字符串的频率、互信息 MI、邻接类别 AV 等统计量来判断该串是否为有意义串，但这些方法很难有效提取低频有意义串。我们发现有意义串倾向于在语料局部集中出现，该特性有利于提取语料中的低频串。如果某个字符串在语料的局部集中出现，即使其频率或 AV 等属性相对比较小，也可以认为该字符串是有意义串。在这个局部区域，该字符串具有相对较高的频率等属性，因此有意义串的这种局部性特征能够提取其他统计量

很难有效提取的低频串。

5.3.1　有意义串的局部性

就时间局部性来说，刚出现的一个有意义串很可能在不久的将来再次出现。图 5.15 和图 5.16 是腾讯论坛语料中的“高考”、“侯耀文”两个词在 2007 年 6 月的时间分布图。从图中可以看出，这两个词都具有明显的时间局部性：“高考”一词的查询数量在高考的 6～8 日达到最高峰，说明在高考期间人们对此事的关注度高；“侯耀文”一词的查询量在这个月的最后几天集中出现，这是“侯耀文逝世”事件引发了该查询词在几天内集中出现的现象。

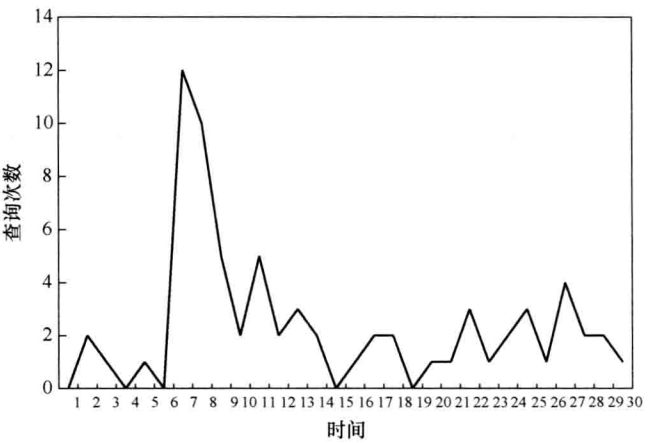


图 5.15　“高考”字符串的时间分布

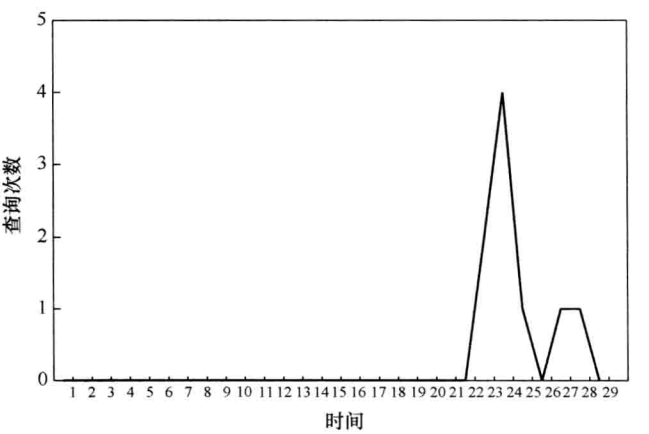


图 5.16　“侯耀文”字符串的时间分布

就空间局部性来说，当一个有意义串在某个位置出现时，很可能在该位置附近再次出现。有意义串的空间局部性反映词语出现的上下文相关性、文档相关性、主题相关性、领域相关性等。例如，奥巴马、萨科齐等新闻人物将在与他相关的国际新闻中出现；“CPU”、“虚拟内存”等很可能在计算机领域的文档中出现；“版主”、“帖子”等词一般出现在 BBS 论坛中。图 5.17 为小说《射雕英雄传》中“天罡北斗阵”、“刘贵妃”、“郭靖”、“黄蓉”四个字符串的位置分布图。前两个字符串只在小说的某几个章节集中出现，体现出该词的局部性；由于后两个字符串是小说的男女主人公，因此这两个字符串出现的频次很高且很难看出其空间局部性。把后两个字符串的位置分布图局部放大后，还是能够看出这两个有意义串空间的局部性分布。

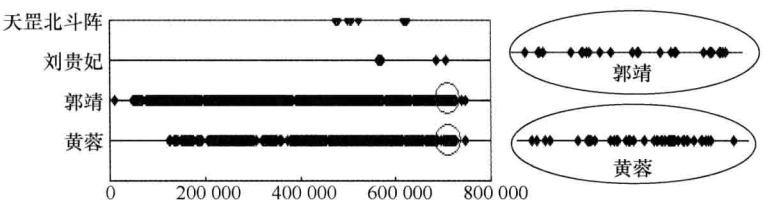


图 5.17 四个字符串的位置分布图

5.3.2 局部性度量

1. 有意义串空间局部性的量化

用 LEspace(S)表示字符串 S 的空间局部性度量值。假设候选字符串 S 在语料中出现 n 次，各出现位置分别为 $P_1, P_2, \cdots, P_n$，则 S 的位置方差 $D(S)$ 为

$$D(S) = \frac{\sum_{i=1}^{n} (P_i - \bar{P})^2}{n-1}$$

其中，$\bar{P}$ 表示字符串的中心位置；字符串 S 的位置方差 $D(S)$ 表示该字符串的各个位置距离中心位置的震荡程度。

位置方差越小，则各个位置距离中心位置越近，各个位置集中程度高，即该字符串的局部性强。$D(S)$ 越小则局部性越好，反之则局部性越差，则字符串的局部性度量为

$$\mathrm{LE}_{\mathrm{space}}(S) = 1/D(S)$$

有意义串在某个区域集中出现时，可能有一两个点远离这个区域出现，则这些点被称为噪声。有些字符串的局部性很好，但由于噪声点的存在，计算出来的位置方差将会很大，该字符串将会被误认为局部性很差。为了更准确地衡量字符串的局部性，应在计算方差之前去掉噪声点。另外，有些字符串会在语料的几个区域集

中出现，形成若干簇，为了更准确地衡量字符串的局部性，也应把这些簇识别出来。为此，利用字符串的位置坐标计算它在任意位置点的密度，根据这些密度值分簇、去噪声。具体步骤如下。

(1) 把语料划分成长度为 1(1 远远小于语料长度)的区域，统计每个区域的字符串个数，把该值当成整个区域的密度。

(2) 根据字符串的密度值找出密度大于一定阈值的连续区间 domain1，domain2，…，domaink，则该字符串集中出现在这些区域。

(3) 对每个区域根据 S 的位置方差公式求位置方差 $D_1(S)$，$D_2(S)$，…，$D_k(S)$。由于要度量有意义串密集出现的程度，因此要选择这些方差中的最小值。根据去噪声点过程后得到结果，可以得到修正后的空间局部性计算方式，即

$$\mathrm{LE}_{\mathrm{space}}(S)=1/\min_{1\leqslant i\leqslant k}\{D_i(S)\}$$

2. 有意义串的时间局部性的量化

将字符串 S 的出现时间按照先后顺序排列，形成时间序列 T_1，T_2，…，T_n。同样利用密度图找到 S 在时间上集中出现的簇，计算每个簇的位置方差，进而求得有意义串的时间局部性度量值 $\mathrm{LE}_{\mathrm{time}}(S)$。如果语料的空间局部性比较明显，如小说语料，则字符串 S 的局部性度量值 $\mathrm{LE}(S)=\mathrm{LE}_{\mathrm{space}}(S)$。如果语料的时间局部性比较明显，如带有时间信息的新闻语料、论坛语料等，则 $\mathrm{LE}(S)=\mathrm{LE}_{\mathrm{time}}(S)$。

5.3.3　算法流程

基于局部性的有意义串提取算法流程如图 5.18 所示。首先是重复串查找阶段，对用户给定的语料 C 分词，从分词结果中选出频率大于一定阈值的字符串作为频繁模式集合 FP(C)(FP(C)中的字符串被称为重复串)。其次是上下文分析阶段，对于 FP(C)集合中的每个重复串 S，计算 S 的 AV 值并过滤掉 AV 值过低的字符串。第三是局部性分析阶段，计算重复串 S 的局部性度量值 LE。最后是综合分析阶段，即根据 AV 值和 LE 值给 S 打分，衡量它能成为有意义串的可能性，如果该值大于一定阈值，则认为该字符串是有意义串。

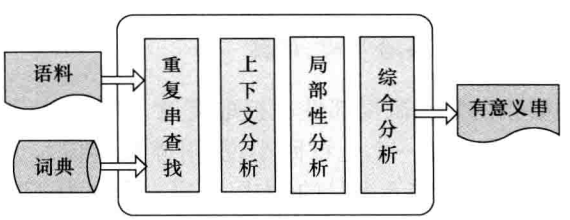


图 5.18　基于局部性原理的低频有意义串生成流程

令 $f(S)$表示字符串 S 成为有意义串的分数值($f(S)$越大,S 越有可能是有意义串)。结合 AV(S)和 LE(S),$f(S)$的计算如下,即

$$f(S)=\lambda\log\left(\frac{\mathrm{LE}(S)+1}{a}\right)+(1-\lambda)\log\left(\frac{\mathrm{AV}(S)}{b}\right)$$

$$a=\max_{S_i\in \mathrm{FP}(C)}\{\mathrm{LE}(S_i)\}+1$$

$$b=\max_{S_i\in \mathrm{FP}(C)}\{\mathrm{AV}(S_i)\}$$

其中,a 和 b 用来归一化;LE(S)有可能为 0,故加 1 用来做平滑;λ 值用来调节 AV 值和 LE 值在 $f(S)$中的权重(λ 越大,LE 值所占的比重越大;$\lambda=0$ 时 $f(S)$完全由 AV 值决定;$\lambda=1$ 时 $f(S)$完全由 LE 值决定)。

如果 S 的 AV 值很大,则 $f(S)$很高,S 会被判定为有意义串;即使 S 的 AV 值不是很大,如果字符串 S 的局部性非常好,LE 值较大,也会被判定为有意义串。因此,用 AV 值和 LE 来给 S 打分是合理的,这样能在不减弱利用 AV 值提取有意义串的效果的同时,用 LE 多获取一些有意义的字符串。基于局部性原理的有意义串提取算法如下所示。

基于局部性原理的有意义串生成算法

输入:语料 C,词典 D,重复串频次阈值θ_1,AV 阈值θ_2,参数 l,参数 λ,密度阈值θ_3,打分阈值θ_4

输出:有意义串集合

1. 对 C 用词典 D 进行分词
2. 对分词后语料 C 发现特征频次大于θ_1的频繁模式,形成集合 FP(C)
3. 对 FP(C)中每个字符串 S 执行步骤 4 到步骤 8
4. 计算 AV(S),如果 AV(S)$<\theta_2$,删除 S
5. 将 C 划分称大小为 1 的区域,计算每个区域的密度值
6. 查找密度大于 θ_3的连续区域 $\mathrm{domain}_1,\mathrm{domain}_2,\cdots,\mathrm{domain}_N$,计算各个区域对应的位置方差 $D_1,D_2,\cdots,D_N$,由 $D_1,D_2,\cdots,D_N$计算 LE(S)
7. 计算 $f(S)$,如果 $f(S)<\theta_4$,删除 S
8. 仍在 FP(C)中的字符串为有意义串,输出有意义串

5.3.4 实验结果与分析

1. 实验目的、实验语料、实验步骤、词典

本次实验的主要目的是测试局部性在提取有意义串的过程中发挥的作用,实验所用的领域语料有两个:第一个是章回小说《射雕英雄传》,大小为 1.7M,包含 42 个章节;第二个为 1998 年前 6 个月的《人民日报》,大小为 27M,包含 102 739 条新闻。章回小说的空间局部性比较明显,基本上每一章叙述一个完整的故事,因此可以用来测试空间局部性原理对有意义串抽取算法的改进效果;新闻语料的时间

局部性比较明显，语料是按照时间顺序组织的，因此可以用来测试时间局部性对有意义串抽取算法的改进效果。除了测试时间局部性的实验是在新闻语料上实现的外，其他所有的实验均在小说语料上实现。实验分为三步。

(1) 调节参数。基于局部性的提取算法中设计了 l 和 λ 两个参数。计算 LE 值需要参数 l，计算 $f(S)$时用到参数 λ。实验这两个参数对准确率的影响。

(2) 测验局部性对低频有意义串提取的改进效果。比较加入局部性和没加局部性的算法对提取低频有意义串的准确率。

(3) 实验比较三种提取有意义串算法(基于 AV 的提取算法、基于 LE 的提取算法、基于局部性原理的提取算法)的效果，即分别按照 AV 值、LE 值、$f(S)$值对重复串集合中的字符串进行排序并取前 N 个结果进行比较。

实验采用的词典大小为 98 346，分词方法为最大正向匹配法。为了减少垃圾串对实验效果的影响，实验中使用停用字表(含“的”、“而”等字)。所有实验均采用人工判别一个字符串是否为有意义串。

2. 参数调节

计算 $f(S)$时用到参数 λ，它用于调节 AV 值和 LE 值在 $f(S)$中的比重。λ 越大，统计量 LE 在 $f(S)$占的比重越大；λ 越小，统计量 AV 所占比重越大。本次实验对 λ 取不同值(范围为 0～1)，每隔 0.1 计算 Top700 的准确率($l=100$)。图 5.19 表明参数 λ 对准确率的影响。从图中可以看出，随着 λ 的增大，准确率先增大再减小，说明结合 AV 和 LE 值可以提高有意义串的查找效率。参数 $\lambda=0.5$ 时，准确率达到了最高点。说明 AV 和 LE 发挥同样作用时，有意义串提取的准确率能达到较好的效果。图 5.20 表明参数 l 对准确率的影响。参数 l 是计算有意义串密度图时的步长。l 过小，很多串将无法找出集中出现的区域；反之，很难去除噪声。实验对 l 取不同值，取值范围为 50～500，每隔 50，计算 Top1000 的准确率。从图中可以看出，随着 l 的增大，准确率先增大再减小。因此，在使用参数 l 时，应选择适当的大小，有利于准确率的提高。

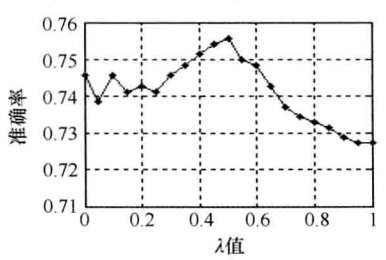


图 5.19　参数 λ 对准确率的影响

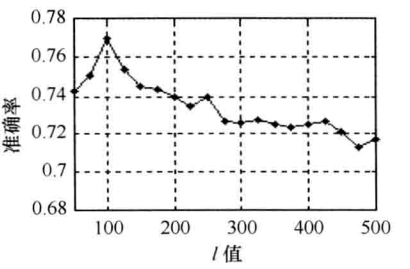


图 5.20　参数 l 对准确率的影响

3. 局部性对低频有意义串提取的影响

表 5.7 比较了基于 AV 方法和基于局部性方法提取低频有意义串的效果。将一定范围内的重复串分别按照 AV 值和 $f(S)$值排序，并计算前 500 个结果的准确率。由于去掉了高频重复串，剩下的重复串的抽取准确率能反应低频有意义串的抽取效率。可以看出，基于局部性原理的抽取方法要明显好于基于 AV 的抽取方法，准确率至少提高了 23.10%。实验表明局部性原理能提高低频有意义串的提取效率。

表 5.7 低频有意义串抽取效果比较

不同方法准确率	频率范围		
	≤20	≤15	≤10
AV	0.580	0.476	0.454
$f(S)$	0.714	0.684	0.634

表 5.8 展示了一些用基于局部性方法提取的低频有意义串。这些有意义串包括人名、地名以及武侠小说专用的词汇。由于较低的频次和 AV 值，这些字符串很难用基于 AV 的抽取方法提取出来。实验结果表明基于局部性的方法能提取有意义的低频字符串。

表 5.8 基于局部性方法提取的低频有意义串

有意义串	频率	AV 值
盖运聪(人名)	8	5
大宛国(地名)	5	4
雅兰花(植物)	5	4
令爱(女儿)	7	4
奇经八脉(专用词汇)	7	5
拙荆(妻子)	5	4

4. 局部性对有意义串提取的影响

图 5.21 比较了三种提取有意义串算法，即分别用 AV、LE、$f(S)$值对有意义串提取的算法的 TopN 准确率。实验中参数 $l=100$，$\lambda=0.5$。从实验结果上看，基于 $f(S)$的方法要明显好于用 AV 排序的方法，说明对空间局部性的度量有利于抽取准确率的提高。

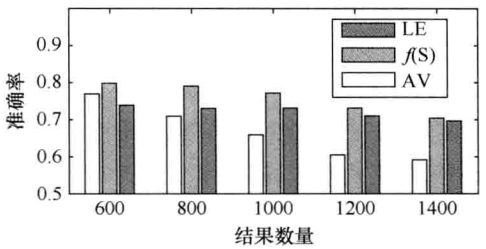


图 5.21　小说语料三种方法准确率比较

图 5.22 的实验结果是对新闻语料抽取有意义串后，对三种算法准确率的比较。从图中可以得到与上图相似的结果，同样可以说明对语料时间局部性的度量也有利于提高抽取有意义串的准确率。

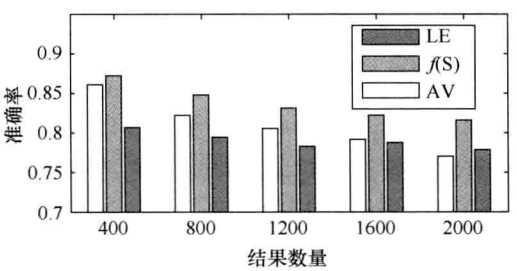


图 5.22　新闻语料三种方法准确率比较

5.4　基于伪相关反馈模型的有意义串提取

采用通用方法提取有意义串时，通常要先对语料进行分词。如果分词时采用的是通用词典，对含有大量新词、新短语的语料分词就可能造成较多的切分错误，进而影响查找有意义串时用到的统计量（如邻接类别等）的数值。而用通用语料计算出的位置成词概率也不一定适用于用户给定的语料。为了解决这个问题，可将有意义串的自动生成过程视为一个信息检索过程，采用检索模型中常用的伪相关反馈模型解决这两个问题。根据用户给定的若干语料，检索出与该语料相关的有意义串集合。与传统信息检索所不同的是，传统信息检索系统一般是根据用户输入的若干查询词，给出若干可能与用户查询词相关的网页的集合。有意义串的生成过程是根据用户提供的特定文档或语料，给出该语料包含的有意义串集合。为此，假设生成候选有意义串的前面若干个结果与给定语料相关，将这些结果放入词典，并用新生成的词典分词，重新发现有意义串。同时，根据用新生成词典的分词结果，更新成词概率。采用该方法能够有效提取给定语料的有意义串，减少垃圾串

的生成。可见，伪相关反馈模型能消除分词对有意义串提取的影响。通过观察有意义串提取结果，可以发现很多垃圾串包含一个相同的字或词。例如，机械专利领域语料中的很多垃圾串包含“包括”这个词，如“装置包括”、“系统包括”、“方法包括”等。这些垃圾串的频次、邻接类别 AV 值等统计特征值都较高，没办法用统计的方法过滤。如果将这些垃圾串包含的相同的字或词查找出来，将有效地删除这些垃圾串。为此，这里提出基于停用字词表的算法，用来删除这类垃圾串，并将该算法作为补充算法加入到基于伪相关反馈模型的提取算法中。

5.4.1 算法的基本思想

首先，选用一个通用词典 D 对用户给定的语料 C 进行分词。根据分词结果更新各汉字的位置成词概率。然后，使用频繁模式发现算法，发现语料 C 的所有特征频次高于一定阈值的频繁模式集合 $\mathrm{FP}(C)_1$。对 $\mathrm{FP}(C)_1$ 的每一个元素 S（S 不在 D 中）计算 S 的位置成词概率，过滤掉位置成词概率低的字符串；计算 S 的 AV 值，将 AV 值过低的字符串滤掉。剩下的字符串形成候选有意义串集合 $\mathrm{CanSet}(C)_1$。对 $\mathrm{CanSet}(C)_1$ 的每个元素 S 计算其相关度 $R(S,C)$，并对候选有意义串集合 $\mathrm{CanSet}(C)_1$ 中的所有元素按照其相关度排序。候选有意义串集合 $\mathrm{CanSet}(C)_1$ 的前 N 个字符串与用户给定语料的相关度高，将这 N 个字符串加入到通用词典 D 中并形成第一个更新词典 D_1。使用 D_1 对语料 C 进行分词，更新各汉字的位置成词概率，发现语料 C 的频繁模式集合 $\mathrm{FP}(C)_2$，同样可以根据新生成的位置成词概率删除垃圾串后得到候选有意义串集合 $\mathrm{CanSet}(C)_2$。同理，可以假设候选有意义串集合 $\mathrm{CanSet}(C)_2$ 的前 N 个字符串是与语料 C 相关的，将这 N 个字符串加入到第一个更新词典 D_1 中并形成第二个更新词典 D_2；可以计算 $D_3, D_4, \cdots, D_n$，直到有意义串集合不再改变，或者没有满足频率和位置成词概率条件的字符串加入到词典。基于伪相关反馈模型的有意义串生成算法流程如图 5.23 所示。

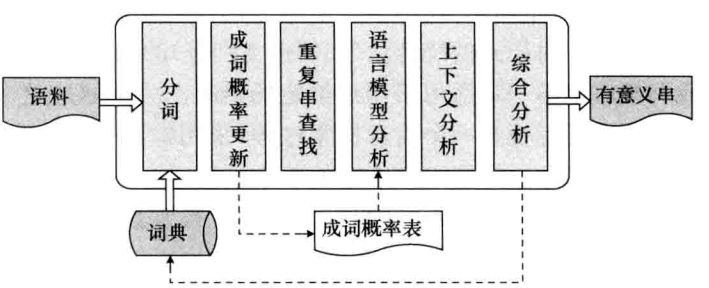


图 5.23 基于伪相关反馈的有意义串生成流程

5.4.2　相关度的定义

给定若干文档构成的语料 C 和字符串 S。字符串 S 对语料 C 的领域相关度，$R(S,C)$一般用 S 在 C 上的特征频次和邻接类别来综合描述。S 在 C 上的特征频次 $f(S,C)$定义为 S 在 C 中出现的次数(用来描述字符串的通用性)；S 在 C 上的邻接类别 $\mathrm{AV}(S,C)$描述字符串的灵活性，定义为 S 在 C 上的左邻接类别和右邻接类别的较小值。相关度 $R(S,C)$定义为

$$R(S,C)=\lambda\times\log(f(S,C))+(1-\lambda)\log(\mathrm{AV}(S,C))$$

其中，$0\leqslant\lambda\leqslant1$ 是一个偏好参数，用于调节领域相关度中特征频次和邻接类别的权重。若 $\lambda=1$，表明领域相关度只考虑特征频次；若 $\lambda=0$，表明领域相关度只考虑邻接类别。

5.4.3　位置成词概率 PWP 的更新

由于用通用语料计算出的位置成词概率不一定适合用户给定的语料，因此这里给出使用迭代生成的新词典来更新位置成词概率表的方法。根据语料 C、词典 D 可以训练得到位置成词概率表。用词典 D 对语料 C 分词，用分词结果计算每个字符 c 在每个位置的成词概率 $\mathrm{PWP}(c,i\mid C,D)$，其中 $i=0$、1、2 分别表示词首、词中、词尾三个位置。根据该成词概率表可以计算出字符串 S 的位置成词概率 $\mathrm{PWP}(S\mid C,D)$。假设 $S=c_1c_2\cdots c_3$，$\mathrm{PWP}(S\mid C,D)$的计算公式为

$$\mathrm{PWP}(S\mid C,D)=\mathrm{PWP}(c_1,0\mid C,D)\times\prod_{i=2}^{n-1}\mathrm{PWP}(c_i,1\mid C,D)\times\mathrm{PWP}(c_n,2\mid C,D)$$

如果 $\mathrm{PWP}(S\mid C,D)$小于一定阈值，则字符串 S 将视为垃圾串滤掉。

使用通用语料 C 计算得到的位置成词概率不一定适合用户给定的语料 C'。我们将语料 C'的部分或全部有意义串加入通用词典 D 形成词典 D'，使用 D'按照同样的分词策略对 C'进行分词，可以统计出 c 在 C'上的位置成词概率，计算出字符串 S 在语料 C'上的成词概率 $\mathrm{PWP}(S\mid C',D')$。

由于通用语料训练得到的位置成词概率训练充分，用户给定语料训练得到的位置成词概率针对性强，因此综合两个训练得到的位置成词概率，字符串 S 在的成词概率 $\mathrm{PWP}(S)$的计算公式为

$$\mathrm{PWP}(S)=\alpha\mathrm{PWP}(S\mid C,D)+(1-\alpha)\mathrm{PWP}(S\mid C',D')$$

5.4.4　算法流程

基于伪相关反馈模型的有意义串自动生成算法如下。

基于伪相关反馈模型的有意义串生成算法

输入：语料 C，通用词典 D_{com}，通用语料得到的各汉字的位置成词概率表 T，重复串频次阈值 θ_1，位置成词概率阈值 θ_2，每次加入词典的有意义串数目 N

输出：语料 C 的领域有意义串集合

1. 初始化 $D=D_{com}$；
2. 对 C 采用词典 D 进行分词，更新 T；
3. 对分词后语料 C 发现特征频次大于 θ_1 的频繁模式，形成集合 FP(C)；
4. 对 FP(C)中每个字符串 S 执行步骤 5 到步骤 7；
5. 计算 S 位置成词概率 PWP(S)；
6. 对若 PWP(S)$<\theta_2$，删除 S；
7. 计算 S 的相关度 $R(S,C)$；
8. 对 FP(C)中的所有字符串按照相关度排序；
9. 将 FP(C)中前 N 的字符串加入 D 中；
10. 若步骤 9 中没有任何字符串加入到 D 中，算法终止；否则，转步骤 2；
11. FP(C)中的字符串为有意义串，输出有意义串。

5.4.5 实验结果及分析

1. 实验环境及实验设计描述

实验采用的是专利语料，包含机械工程、照明、采暖、武器、爆破方向的专利申请说明共计 2859 条，语料大小为 1178M。专利语料领域性强，包含大量的专业术语，适合用于检验本章提出的算法的有效性。实验采用的通用词典大小为29 803，分词方法为最大正向匹配法。实验中使用的通用位置成词概率表是由 1998 年的《人民日报》语料训练得到的。

实验的目的是验证基于伪相关反馈模型算法的有效性、停用字词表的有效性以及两者结合的有效性。实验分成两大部分：第一部分测试基于伪相关反馈模型算法的有效性，首先测试随着迭代次数的曾加提取结果的 P、R 和 F 值的变化，然后用实验中的具体例子，解释 P 和 R 值增加的原因；第二部分测试停用字词表的有效性，首先测试加入停用字词表后 P、R 和 F 值的变化，然后测试两种算法结合后 P、R 和 F 值的变化。

2. 实验结果及分析

(1) 基于伪相关反馈模型算法的提取实验

实验目的是测试基于伪相关反馈的有意义串生成算法的效果。用该算法提取有意义串，随着迭代次数的曾加，比较提取结果的 P、R 和 F 值。本次实验用 AV$>$4和 PWP$>$0.01 为经验阈值生成有意义串，以 AV 值将有意义串排序，将前

5%的有意义串加入到词典中，生成新的词典。实验结果如表 5.9 所示。

表 5.9　采用迭代方法抽取有意义串的效果

迭代次数	P	R	F 值	MS 数
0	0.670522	0.841714	0.746428	3545
1	0.708853	0.844901	0.770921	3366
2	0.726691	0.844547	0.781199	3282
3	0.739658	0.842068	0.787548	3215
4	0.746137	0.837819	0.789324	3171

从表 5.9 中可以看出，迭代结果的 P 值、F 值要比没有迭代结果的高。第四次迭代结果的 F 值提高了 5.8%，P 值提高了 13.1%。单是第一次迭代的结果，F 值就提高了 4.0%，P 值提高了 5.8%。并且第 1 次迭代结果的 R 值也比没有迭代的结果提高了 0.4%。从数据分析的结果可以看出，基于伪相关反馈模型的提取算法效果要好于没有伪相关反馈算法的结果。

从图 5.24 可以看出，随着迭代次数的增加，P 值不断变大，R 值和 F 值先变大再变小。P 值不断变大说明迭代方法能够有效地删除垃圾串。通过迭代的方法，一些新的有意义串可以被抽取出来，因此 R 值先增大。F 值在第九次迭代的时候数值达到最高峰。实验结果表明，随着迭代次数不断曾加，P 值将不断增加；选择适当的迭代次数，F 值将达到其峰值。该结果说明伪相关反馈模型能提高有意义串提取的 P 值和 F 值。

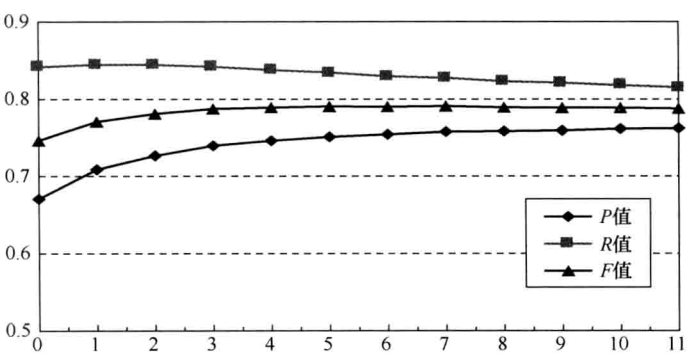


图 5.24　采用迭代方法抽取有意义串的效果

准确率的提高是因为向词典中加入领域有意义串使得分词结果更准确，减少了垃圾串的产生。例如，没有迭代的方法中有“发动机控制”这个垃圾串。该串出现了 19 次，AV 值为 7，其右邻接集合＝{单元、设备、信号、装置、系统、技术、变量}。第一次迭代时，加入词典的词中包含“控制单元”、“控制信号”、“控制装置”、

“控制系统”。通过新一轮的有意义串查找,“发动机控制”出现频次为 5,AV 值为 3,低于规定的阈值,因此被当做是垃圾串滤掉。类似的垃圾串还有“继电器控制”、“热装置”、“阀连接”等,第一次迭代以后,删除这样的垃圾串 232 个,因此准确率提高了 5.8%。

由于有意义串的加入,使得有些字符串的上下文环境更加丰富,AV 值大于给定的阈值,所以能成为有意义串。因此,相对于无迭代的方法,迭代的方法生成了若干新有意义串。表 5.10 是新有意义串的例子。

表 5.10 迭代方法召回的有意义串

	新生成的有意义串
2-gram	模数、管轴、套置
3-gram	作动器、联轴节、层压板、柔滑油
4-gram	水平框条、定位支架、风油比例
其他	真空集热器、热声制冷机、家用中央空调

通过迭代的方法能够多召回一些有意义串,提高了基于伪相关反馈模型方法的召回率。

(2) 加入停用字词的实验效果

本次实验的主要目的是测试停用字词对有意义串提取的改进效果。首先要构建停用词表和位置停用字表,将词典词分别按照 AV 值和频率排序,各取前 50 个结果,针对这些词进行人工判断,将停用字词加入相应的表中。构建过程结束后,加入到停用词表的词集合={通过、进行、并、形成、包括、安装},加入位置停用字表的字集合={板、式、部、器、型、使、时、本}。

首先单独使用停用字词表提取有意义串。从表 5.11 可以看出加入人工判断的停用词和位置停用字后,有意义串提取结果的改进程度。SW(stop word)代表加入人工判断的停用词的结果,PSC(position stop char)代表加入人工判断的位置停用字的结果。

表 5.11 加入人工判断停用词的效果

	P	*R*	*F* 值
无人工判断	0.670522	0.841714	0.746428
SW	0.697442	0.839943	0.762088
PSC	0.691835	0.837110	0.757571
SW+PSC	0.718331	0.835340	0.772430
改进效果/%	7.2	−0.7	3.5

通过加入 6 个停用词和 8 个位置停用字,*P* 值提高了 7.2%,*R* 值降低了

0.7%，F 值提高了 3.5%，说明加入的字词能够有效地删除大量垃圾串，召回率上有些损失，但 F 值是有提高的。用较少的人工使得查找结果的准确率提高了 7.2%，说明人工判断停用字词的方法能够改善有意义串提取的效果。

将基于伪相关反馈原理的算法和基于停用字词的算法结合在一起提取有意义串。实验结果如表 5.12 所示。

表 5.12　结合算法提取有意义串的效果

	P	R	F 值
没有迭代	0.670522	0.841714	0.746428
有迭代	0.746137	0.837819	0.789324
有迭代＋停用字词	0.783883	0.833569	0.807963
改进效果/%	16.9	−1.0	8.2

可以看出，两种方法结合的准确率提高了 16.9%，F 值提高了 8.2%，说明两种方法结合能有效地提取有意义串。迭代的方法和停用字方法结合要比单独使用迭代方法的提取效果要好，说明停用字词表能有效地提高迭代方法的准确率。

5.5　本章小结

有意义串提取中用到重复串发现算法、上下文邻接分析方法、双字耦合度和单字位置成词概率等，采用 N 元递增方法发现重复串。上下文邻接分析通过计算字符串的邻接种类、邻接熵、邻接对、邻接对熵等邻接特征量，反映字符串在真实文本中的语用灵活性。用双字耦合度来描述字与字之间结合的紧密程度，将其与单字位置成词概率相结合来对字符串的语义完整性做智能分析。本章分析了有意义串的空间局部性和时间局部性，提出了结合 AV 值和 LE 值挖掘有意义串的方法，并用实验证明局部性原理的有效性，提出基于伪相关反馈模型的有意义串提取算法，解决了分词错误以及位置成词概率不适用对有意义串查找造成的影响，并给出基于停用字词表的提取算法。通过不同的实验，验证了此算法对于处理大规模的网络文本切实可行，而且得到较高的准确率。

相关工作与扩展阅读

新特征词和有意义串的提取工作将在重复串发现的基础上进行。重复串是指在一个字符串中出现次数超过两次的子串。重复串发现就是在序列 $w=c_1c_2c_3\cdots c_n$ 中进行串频统计，得到所有的频度大于预设阈值的重复串和它们的频度（胡吉祥，2006；邹刚等，2004）。重复串发现在有意义串提取的整个过程中处于比

较重要的地位，已经应用于中文文本的重复串查找算法有后缀树算法、sequitur 算法和 n 元递增分步算法等(邹刚等，2004)。通过后缀树算法来发现重复串的方法，适用于小规模语料；Sequitur 算法的时间复杂度和空间复杂度都是线性的，但查找过程会漏掉某些特殊字串；n 元递增分步算法虽然时间复杂度比较高，但是空间复杂度较低。在文献(郑家恒等，2002)对于网络新词语的研究中，利用 n 元递增分步算法来发现重复串，其主要思想是先统计所有两字串的频次，然后再逐步统计三字串、四字串、五字串…记录每一次扩展的字串以及对应频次，到达句末或者是字串长度达到阈值时停止扩展。

有很多统计量是用来衡量字符串内部紧密连接程度的。字符串内部的字或词连接的越紧密，就说明该字符串越有可能是一个有意义串的单元。最常见的衡量内部紧密连接性的统计量为 MI 和 log-likelihoo。文献(王开铸等，1995)统计汉字的共现概率并以此作为依据判断分词碎片是否构成词。文献(Chien，1997)用 PAT-Tree 数据结构存储所有在文中出现的字符串，统计其出现频次并把这些字符串作为候选的关键模式，用基于 MI 的过滤策略过滤掉 PAT-Tree 中不完整的字符串。文献(Weeber et al.，2000)用 log-likelihood 和 Fisher 抽取检测来提取所有频率的新词和短语。文献(Evert et al.，2001)用 MI、log-likelihood ratio test、t-test、χ^2-test 和共现频率等统计量衡量词内部元素的紧密连接程度。文献(Lai et al.，2002)用 PLR(phrase-like unit likelihood ratio)来衡量字符串能够成为一个词的概率，从整个语料中挑选 PLR 大于一定阈值的 N-gram 作为候选串，删除含有停用字和交叉歧义的字符串并得到最终结果。文献(Piao et al.，2006)用 log-likelihood 和 t-score 两个统计量从 N-gram 中选择候选串，经过词性模式过滤得到有意义串，即文献(Zhang et al.，2007)用快速的算法提取语料中重复出现的字符串，并用 MI 判断抽取出来的字符串是否为有意义串。

基于统计模型的方法是利用各种统计模型提取语料中的有意义串。文献(Frank et al.，1999)提取文档中的 N-gram，除去含有停用字的字符串并除去不含名词的字符串，剩下的作为候选串，根据候选串 key 的 T(tf×idf)以及第一次出现的位置 D，用 Bayes 方法求 key 的出现概率 Pr(key$|T,D$)。文献(Turney，2000)用C 4.5作为分类器提取有意义串。文献(Li，2004)主要研究了两种类型的有意义串，即 NW11(由两个汉字组成)和 NW21(由一个二字词后跟一个单字组成)。作者将新词识别问题看作一个二元分类问题，对 NW11 和 NW21 分别选用不同的特征并利用 SVM 分类器进行分类。除用分类模型外，语言模型、CRF 模型、决策树模型、隐马尔可夫模型(Sun et al.，2008)、MIIM 模型、最大熵模型等也被应用到有意义串提取工作中。文献(Tomokiyo et al.，2003)认为有意义串要具备两个特征，即成词度和信息度，通过不同语言模型的 pointwise-KL 距离来描述成词度和信息度，用前景语料的 N 元语言模型与一元语言模型的 pointwise-KL 衡量成词

度，用前景语料与背景语料的 N 元语言模型 pointwise-KL 衡量信息度，最后将成词度和信息度加权平均，用得到值对候选字符串进行排序。文献(Peng，2004)完全采用统计的方法，将分词和有意义串发现作为一个统一的过程，利用 CRF(conditional random fields)模型，将词汇特征和领域知识等很好地融入到模型中，同时发现的有意义串会加入到词典中改善分词器的性能，改善后的分词系统又能更好的识别新词，两者互相促进，达到了很好地识别效果。文献(Fuxman et al.，2008)将提取领域关键词的任务当成一个标注问题，初始给定一些领域以及与该领域相关的 Qurey 和 URL。在 query 和 url 构建的图上，用随机游走方法把所有的 query 和 url 都标注上领域信息，并且建立一个统一的框架，用马尔可夫条件随机场来解决这个问题。邹纲等(2004)提出了一种自动检测新词语的方法，通过大规模地分析从 Internet 上采集而来的网页，建立巨大的词和字串的集合，从中自动检测新词语，而后再根据构词规则对自动检测的结果进行进一步的过滤，最终抽取出采集语料中存在的新词语。根据该方法实现的系统，可以寻找不限长度和不限领域的新词语。邓擘等(2006)指出基于 Tf-Idf 的常用文本特征提取方法在文本分类问题中的缺陷，进而提出使用特征词的分布状态、词频和文本频三者相结合的方式提取文本特征的观点，给出了计算特征词权重的新方法，提出了新的文本分类方法，方法能够最大限度保留文本的特征，并且可以有效避免向量空间模型中的维数灾难问题。贺敏等(2007)提出了一种基于大规模语料的新词识别方法，在重复串统计的基础上，结合分析不同串的外部环境和内部构成，依次判断上下文邻接种类，首尾单字位置成词概率以及双字糖合度等语言特征，分别过滤得到新词。黄玉兰等(2009)提出基于局部性原理的有意义串提取算法。作者发现有意义串遵循空间局部性原理和时间局部性原理，并用 LE 值量化字符串的局部性。在此基础上，利用 LE 值和其他统计共同提取有意义串。崔世起等(2006)提出了一种基于大规模语料库的新词检测方法，首先在大规模的 Internet 生预料上进行中文司法切分，然后在分词的基础上进行频度统计得到大量的候选新词，针对二元新词、三元新词、四元新词等常见的模式，用自学习的方法产生 3 个垃圾词典和一个词缀词典对候选新词进行垃圾过滤，最后使用词性过滤规则和独立词概率技术进一步过滤，据此实现了一个基于 Internet 的进行在线新词检测的系统，并取得了令人满意的性能。

参 考 文 献

崔世起，刘群，孟遥，等．2006. 基于大规模语料库的新词检测．计算机研究与发展，43(5)：927-932.

邓擘，樊孝忠，杨立公．2006. 基于统计分布与集合论的文本分类方法．北京理工大学学报，26(7)：589-593，597.

贺敏，龚才春，张华平，等．2007．一种基于大规模语料的新词识别方法．计算机工程与应用，43(21)：157-159.

胡吉祥．2006．基于频繁模式的消息文本聚类研究．中国科学院博士学位论文.

黄玉兰，龚才春，许洪波，等．2009．基于局部性原理的有意义串提取方法// 第四届全国信息检索与内容安全学术会议论文集.

王开铸，李俊杰，吴岩．1995．无词典自动分词的研究．计算语言学进展与应用，13(2)：35-38.

王小飞．2006．面向大规模信息检索的中文分词技术研究．中国科学院博士学位论文.

郑家恒，李文花．2002．基于构词法的网络新词自动识别初探．山西大学学报(自然科学版)，25(2)：115-119.

邹刚，刘群．2004．面向 Internet 的中文新词语检测．中文信息学报，18(6)：1-9.

Chien L F. 1997. PAT-tree-based keyword extraction for Chinese information retrieval// Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.

Evert S, Brigitte K. 2001. Methods for the qualitative evaluation of lexical association measures// Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics.

Frank E, Paynter G, Witten I, et al. 1999. Domain-specific key phrase detection// Proceedings of IJCAI: 668-673.

Fuxman A, Panayiotis T, Kannan A. 2008. Using the wisdom of the crowds for keyword generation// Proceedings of the 17th International World Wide Web Conference.

Lai Y S, Wu C H. 2002. Meaningful term extraction and discriminative term selection in text categorization via unknown-word methodology. ACM Transactions on Asian Language Information Processing, 1(1): 34-64.

Li H Q. 2004. The use of SVM for Chinese new word identification// Proceedings of First International Joint Conference on Natural Language Processing.

Peng F C, Feng F F, Andrew M. 2004. Chinese segmentation and new word detection using conditional random fields// Proceedings of the 20th International Conference on Computational Linguistics, Switzerland.

Piao S S L, Sun G F, Paul R, et al. 2006. Automatic extraction of Chinese multiword expression with a statistical tool// Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics.

Sun M S, Xu D L. 2008. Disyllabic Chinese word extraction based on character thesaurus and semantic constraints in word-formation// Proceedings of the 11th International Conference on Text, Speech and Dialogue.

Tomokiyo T, Hurst M. 2003. A language model approach to key phrase extraction// Proceedings of the 41th Annual Meeting of the Association for Computational Linguistics.

Turney P. 2000. Learning algorithms for key phrase extraction. Information Retrieval, 2(4): 303-336.

Weeber M, Rein V, Baayen R H. 2000. Extracting the lowest frequency words: pitfalls and pos-

sibilities. Computational Linguistics, 26(3): 301-317.

Wu A D, Jiang Z X. 2000. Statistically-enhanced new word identification in a rule-based Chinese system// Proceedings of the Second Chinese Language Processing Workshop.

Zhang Y, Liu C H. 2007. An improved fast algorithm of frequent string extracting with no thesaurus// Proceedings of the 6th International Conference on Artificial Intelligence.

第 6 章　大数据聚类与分类

6.1　概　　述

随着互联网的发展，搜索引擎扮演着越来越重要的角色。然而，当前搜索引擎返回的结果往往内容庞杂，常常缺乏有效地加工整理，一般不能满足用户需求。究其原因，首先，海量的搜索结果以庞大的线性列表的方式组织，用户只能顺序地浏览并查找所需内容，效率低下；其次，搜索结果歧义混杂，同一搜索词的结果往往有多种形式，搜索词本身也存在歧义。例如，“苹果”部分搜索结果和水果相关，另一部分是“苹果”公司的相关动态，搜索引擎将这些内容杂乱得混合到一张列表中返回给搜索该词的用户，用户需要耗费大量的精力才能筛选出想要的信息。此外，绝大多数普通用户往往很难采用精确的搜索词清晰地表达自己的信息需求，而搜索引擎设计的前提往往假设是用户的查询是精确的，对搜索词的潜在语义扩展（类似于联想功能）是极其必要的。

聚类是按照事物的某些属性和特征把事物聚集成类，使类间的相似性尽量小，类内的相似性尽量大。互联网中新出现的大多数有意义串都是随着某个新事件、新话题出现的。例如，随着超级女声的火爆出现了新词语，如“超女”、“海选”、“PK”、“躺枪”等。从另一个角度来分析，给出几个关键词语，就可以确定一个话题或者事件。搜索结果聚类将不同意义的内容以语义聚团的方式组织，解决歧义混杂的问题，方便用户查看信息。搜索结果聚类对查询结果进行全面的分析处理，对同类的结果归类，全面地阐释查询对象，类别标签可以给出查询对象的一组扩展关键词。这些关键词不仅体现了查询对象的特点，而且可以帮助构建新的查询，以找到更全面准确的信息。例如，查询“美国总统大选”，如果聚类标签是“奥巴马”、“麦凯恩”，用户就可以猜测出这是和总统大选有关的两个人，从而获得该事件的一个重要信息。可见，搜索结果聚类能较好地解决传统搜索引擎呈现搜索结果的诸多不足。

关键词提取，即计算词汇、短语与文章主题内容相关度，按相关度排序返回最能表现文章主题的前 N 个词汇或者短语。针对海量的文档信息，人们提出了很多关键词自动提取方法，主要分为基于统计特征的方法（如词语频率统计）、基于词语网络的方法（即根据一定规则将文档映射为词语网络，利用词语网络计算词语的关键度）、基于语义的方法（即利用词语的语义特征提取关键词）。本章给出基于关键词提取的聚类。首先，通过关键词之间的关系可以发现文章之间内在的联系；其

次，关键词提取过程中使用多特征融合的评价方法，即使文章较短，也能提取出高质量的关键词；最后，提取过程中利用互信息计算关键词与搜索词的相关性，并且采用一定的算法防止针对搜索引擎的内容作弊，使得关键词提取方法更适用于检索结果聚类的特征选择。因此，基于关键词提取的检索结果聚类能够得到更好的聚类效果。

6.2　基于关键词提取的搜索结果聚类

6.2.1　相关术语简介

定义相关术语如下。

(1) K：用户输入的搜索关键词。

(2) $D=\{d_i\}$：D 为搜索结果集合，其中 d_i 是由搜索结果标题与摘要组成的第 i 个文档。由于正文的获取速度较慢，不能满足在线聚类的要求，所以这里使用摘要与标题替代全文进行聚类。

(3) $C=\{\mathrm{DT}_1, \mathrm{DT}_2, \cdots, \mathrm{DT}_i, \cdots, \mathrm{DT}_n\}$：聚类结果集合，其中 DT_i 为第 i 个类别的文档集合。

(4) TD = tag_of(D)：关键词提取的形式化定义，TD 为关键词集合。

(5) $C=$ KBSC (K, D)：KBSC(keyword based search clustering)聚类算法的形式化定义。

聚类过程首先对集合 D 进行关键词提取得到关键词集合 TD，即 TD = tag_of(D)；其次以关键词作为基础特征进行聚类，即 $C=$ KBSC (K, D)。

6.2.2　关键词提取

1. 关键词提取

1) 候选词提取

使用前面提到的 ICTCLAS 分词程序对搜索引擎返回的元信息进行分词。对分词后的结果去停用，过滤一些常见的停用词。例如，“是”、“的”(新版的 ICTCLAS 提供专业词典与用户词典的功能，用户可以添加自己的词典)。我们在分词的过程中加入了关键词词典。该词典中收录了大量人工标引的关键词。一些期刊、杂志网站收录了许多人工标注的关键词的文章，并且这些文章都有具体的分类，从学术文章到科普生活类的文章，包含的词语非常丰富。这里使用的关键词词典就是这些文章中的关键词。首先，使用采集程序抓取包含这些文章的网页，并抽取其中的关键词、类别等信息，获得 200 万左右词汇的词典(该词典中存在部分垃圾词汇，主要是抽取过程中带来的噪声，以及部分期刊的关键词是机器自动生成的垃圾词汇)。然后，对词典进行人工清洗，从而获得最终 170 万的关键词典。

2）新词发现

基于词典的关键词最大的不足之处就是不能发现新词，而新词往往包含重要的信息，很可能是当前的热点。例如，搜索“流感”，当前的热点是“禽流感”，词典中未收录该词，则这样重要的信息就会丢失。文献（黄玉兰，2008）提出的基于局部性原理的有意义串提取方法对新词的识别有较好的效果，能够识别出“禽流感”这样的词，是解决新词发现的有效方法。但是该方法应用在检索结果聚类中会带来新的问题。例如，对于给定的语料，使用该方法可能同时识别出“禽流感”、“禽流感疫情”、“禽流感病毒”等词。这些词往往词义相近或者相同，如果都作为类别的名称，会给类名选择的过程带来新的噪声，使聚类效果下降。各类间不仅意义相近，而且类别中的文档重复率高。本章针对上述问题，对该算法进行改进，加入词语相似性的计算，合并意义相似的词语，使有意串的方法更适用于检索结果聚类。改进后的算法如下，其中 $f(S)$ 表示 S 成为有意义串的分数值，$f(S)$ 值越大，S 越有可能是有意义串。

基于局部性原理的有意义串生成算法改进

输入：语料 C，词典 D，重复串频次阈值 θ_1，邻接类别 AV 阈值 θ_2，密度阈值 θ_3，打分阈值 θ_4，$W(s)$ 表示 s 的字符集合，相似度阈值 θ_5

输出：有意义串集合

1. 对 C 用词典 D 进行分词
2. 对分词后语料 C 发现特征频次大于 θ_1 的频繁模式，形成集合 FP(C)
3. 对 FP(C) 中每个字符串 S 执行步骤 4 到步骤 9
4. 计算 AV(S)，如果 AV(S)$<\theta_2$，删除 S
5. 将 C 划分称大小为 1 的区域，计算每个区域的密度值
6. 查找密度大于 θ_3 的连续区域 domain_1，domain_2，…，domain_N，计算各个区域对应的位置方差 D_1，D_2，…，D_N，由 D_1，D_2，…，D_N 计算局部性度量值 LE(S)
7. 计算 $f(S)$，如果 $f(S)<\theta_4$，删除 S
8. $\forall S_i \in \text{FP}(C)$，如果 $W(s) \cap W(s_i) > \theta_5$，删除 S
9. 仍在 FP(C)中的字符串为有意义串，输出有意义串

2. 多特征融合的关键词评价方法

1）TF 特征

TF 表示词语在文档中出现的频率（词频）。一般认为一个词的 TF 越高，则该词在这篇文章中越重要。但是搜索引擎返回的元信息不同于一般性的文章，通常的搜索引擎使用采集器爬取互联网上的信息，不仅收录了有用的信息，也抓取了大

量垃圾信息。例如，一些网站为了进行搜索引擎优化，在一个页面中包含大量的重复的词。当用户搜索该词时就会返回该页的信息，而这样的信息对用户是没有用的。令文章的词总数为 C_{total}，$P_i = \text{TF}/C_{\text{total}}$，当 $P_i > P$ 时（P 为阈值，一般取值 0.75），则认为该文章包含垃圾信息过多，文章重要度低。

$$\text{TF}_{\text{new}} = \begin{cases} \text{TF}, & \text{TF}/C_{\text{total}} > P \\ 0, & \text{TF}/C_{\text{total}} \leqslant P \end{cases}$$

2）词性特征

我们发现关键词一般都是名词（n）与动词（v），还包含少量的形容词与副词，介词和助词等一般不能表述具体的意义。此外，名词中的人名（nr）、地名（nt），机构名（ns）等更有可能成为关键词。所以本章使用词性进行关键词权重的调整。

令集合 $A=\{\text{nr}, \text{nt}, \text{ns}, \text{v}\}$，$t$ 为关键词，P_{weight} 为词的词性权重，p 为候选关键词词性，T 关键词集合，P 为关键词对应词性权重集合，a、b 和 c 为可调节变量，一般取值分别为 3、2 和 1。如果 $\forall p \in A$，当 $p=\text{nr}$ 时，$P_{\text{weight}} = \text{TF}_{\text{new}} * a$；当 $p=(\text{ns}|\text{nt})$ 时，$P_{\text{weight}} = \text{TF}_{\text{new}} * b$；当 $p \in n \cap p \notin A$ 或者 $p=\text{v}$ 时，$P_{\text{weight}} = \text{TF}_{\text{new}} * c$，则将 t 加入集合 T 中，将 P_{weight} 加入集合 P 中。最终集合 T 为过滤后的关键词集合，P 为对应词的词性权重。

3）互信息特征

检索词是检索结果的核心主题，在检索词周围出现的词更有可能成为关键词。例如，搜索“流感”，在检索结果中“流感”周围出现的词频较高的是“H7N9”、“疑似病例”等。这些词都能较好的体现搜索结果的主题，更适合做关键词，所以关键词的权重大小应该体现检索词与关键词之间的关系。这里使用互信息表示两个词的依赖关系。根据文献（Zhao，2006），互信息定义为

$$\text{MI} = \text{Log}\frac{p(\text{query}, \text{term})}{p(\text{query}) * p(\text{term})}$$

其中，分子 $p(\text{query}, \text{term})$ 表示关键词与查询词在一句话中的共现频率；分母 $p(\text{query})$ 和 $p(\text{term})$ 分别表示查询词和关键词分别出现的频率。

4）多特征融合

这里根据 TF、词性和互信息三种特征在关键词中评价中重要程度的不同，给出下列公式，即

$$w_i = k_1 \text{TF}_{\text{new}} + k_2 P_{\text{weight}} + k_3 \text{MI}$$

其中，k_i 为可调节参数，一般取值为 0、1 和 3。

在关键词抽取过程中，通过将 TF 特征、词性特征及互信息特征融合到统一模型的多特征融合模型进行评价。

6.2.3　基于关键词的检索结果聚类方法

搜索引擎为使用户能够得到更好的用户体验，检索结果一般按照相关度进行

排序，越相关的文档越排名靠前。所以，Top-N 的文档能够代表用户检索的意图，包含大部分用户希望检索文档的类型。考虑到检索结果聚类的实时性要求，使用 Top-N 文档进行聚类，分析获取类名，使用分类方法，将文档划分到已有的类别中。基于上述思想，提出基于关键词的检索结果聚类方法 KBSC。

检索结果聚类 KBSC

输入：检索词 K，文档集合 D

输出：类集合 DT

1. 从集合 D 中选取 Top-N 文档形成文档集合 DD
2. 对 $\forall d_i \in D$ 执行 tag_of(d_i)得到集合 $T\{TD_1, TD_2, \cdots, TD_i, TD_n\}$，其中 $TD_i = \{t_1, t_2, \cdots, t_j, t_n\}$ 表示文档 d_i 的关键词集合，t_i 为第 i 个关键词
3. 对于 $t_j \in TD_i (0 < i < |T|, 0 < j < |TD_i|)$，如果 $\exists t_k \in TDD$，t_k 的值与 t_j 相等，则 t_k · weight $+= t_j$ · weight，否则将 t_j 加入集合 TDD 中
4. 对于每一个 $t_i \in TDD (0 < i < |TDD|)$，如果 $t_i \in d_i$，则将文档 d_i 加入集合 DT_i 中，形成集合 $DT = \{DT_1, DT_2, \cdots, DT_i, \cdots, DT_n\}$
5. 对于所有 $0 < i < |DT_i|, 0 < j < DT_j$，如果 $DT_i \cap DT_j / \min(|DT_i|, |DT_j|) > p_{dt} (p_{dt} = 0.75)$，执行 6；如果 $DT_i \cap DT_j / \max(|DT_i|, |DT_j|) > p_{mdt} (p_{mdt} = 0.9)$，执行 7，否则执行 8
6. 将 t_i 与 t_j 合并，形成新的 t_i，删除集合 DT_j
7. $DT_{min} = DT_i$，$DT_{max} = DT_j$；如果 $|DT_i| < |DT_j|$，则 $DT_{min} = DT_j$，$DT_{max} = DT_i$；DT_{min} 为 DT_{max} 的子类别
8. 从集合 DT 中选取 TopN 形成集合 $C = \{DT_1, DT_2, \cdots, DT_i, \cdots, DT_n\}$ 为最终类别

6.2.4 实验结果及分析

1. 实验设计

这里采用用户评价的方法，对类名进行评分，使用 30 个不同类型的查询，如表 6.1 所示，查询的类型包括歧义查询词、热点词和一般关键词。热点词使用搜索引擎中热门搜索词中的前 10 个词，这些词的类型丰富，包含新闻、娱乐、军事和体育等。

表 6.1 不同类型关键词列表

类型	举例
歧义查询词	苹果，sun，甲壳虫，…
热点词	开心网，我的兄弟叫顺溜，NBA，…
一般关键词	总统大选，世界杯，信息检索，…

这里使用百度搜索引擎作为数据源，每个查询词采集 200 条搜索结果进行聚类。

2. 多特征融合方法评测

采用表 6.1 构造的查询词对多特征融合方法进行测试，测试词性 POS 与互信息 MI 对多特征融合公式的影响，希望通过实验验证多特征融合公式的有效性，从表 6.2 可以看出，词性对准确率的影响较大，因为词性能够过滤介词和助词等一般不能表述具体意义的高频词汇，调整名词和动词的权重；互信息对准确率的影响较小。但是，互信息能够发现重要的低频类名，例如搜索“总统大选”，搜索结果大多数都是与“美国总统大选”相关内容，“伊朗总统大选”、“俄罗斯总统大选”等信息较少、排名靠后。这些词语与关键词之间的互信息值较大，通过互信息特征能够将这些类别的排名提前，提高用户满意度。因此，多特征融合的方法能够更好地描述关键词，获得更高准确率。

表 6.2　不同特征排序算法评价

算法	P@3	P@5	P@7	P@10
TF+MI/%	77	80	73	68
TF+POS/%	83	84	82	79
TF+POS+MI/%	86	86	84	80

3. 与已有系统对比实验

使用表 6.1 构造查询词，将这里提出的聚类方法产生的聚类结果与 vivisimo 聚类结果进行比较。vivisimo 是一个元搜索引擎，以层次化聚类的形式展现搜索结果，是国际上公认的搜索结果聚类软件。测试对象是中国科学院 40 位不同专业的学生、老师，其中有 65%的人认为 KBSC 的聚类结果优于 vivisimo，35%人认为 vivisimo 聚类效果更佳(图 6.1)。图 6.2 是用关键词“总统大选”进行的对比，左边为 KBSC 聚类结果，右边是 vivisimo 聚类结果。

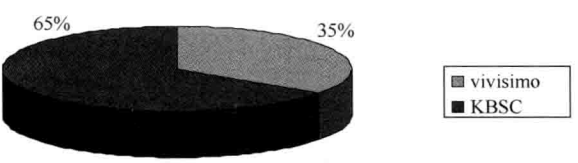


图 6.1　KBSC 与 vivisimo 对比

对 KBSC 产生的聚类结果进行了用户满意度调查结果如图 6.3 所示，其中 55%的用户对聚类结果较为满意，30%的用户认为聚类效果一般，15%的认为聚类

效果较差。据调查聚类结果存在的主要问题是类别的重复。在图 6.2 中，类别“美国总统大选”子类别中又包含了“选举”。从图 6.2 可以看出，vivisimo 也存在同样问题。



(a) KBSC (b) vivisimo

图 6.2 关键词“总统大选”聚类结果

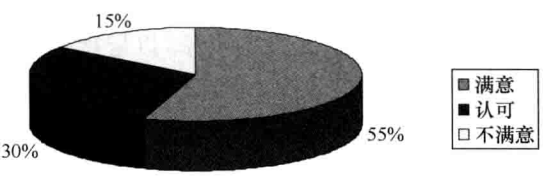


图 6.3 KBSC 用户满意度调查

6.3 基于 K-means 算法的有意义串主题聚类算法

关于某个特定主题的有意义串经常会出现在与该主题相关的文档中，由此可以作出推测，经常在同一文档中出现的有意义串应该是与同一主题相关的。如果将文档作为特征对有意义串进行聚类，经常在同一文档中出现的串应该聚成一类，希望得到的结果是一个个关于某个特定主题的有意义串的集合。也就是说，将有意义串作为聚类点，而出现这些串的文档作为聚类特征，有意义串主题的主要步骤

跟文本聚类基本相似。它们的对比如图 6.4 所示。

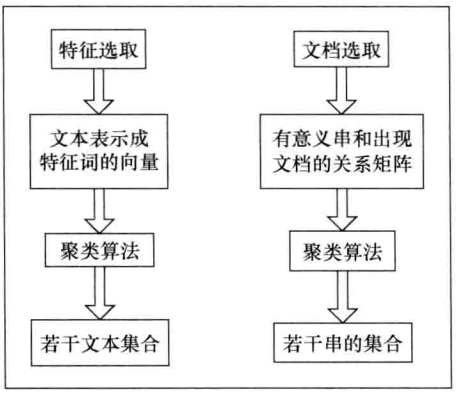


图 6.4　文本聚类与有意义串聚类对比图

图 6.4 左侧是文本聚类过程，首先将文本表示成机器可计算的形式，可以是抽取文本特征形成一个向量或者其他特殊的结构，然后选择某种聚类算法进行聚类并得到若干个文档集合，每个集合内部的相似度较高。图 6.4 右侧显示了有意义串的聚类过程，首先选择一些对于聚类区分度大的文档作为特征，然后将有意义串和出现文档的关系表示成向量，再进行聚类，得到若干主题相关的有意义串的集合。

在第 5 章对有意义串提取时介绍了 n 元递增算法查找重复串，这种方法记录了每个串的文档链和邻接链等信息。这些信息虽然对于有意义串提取没有直接贡献，但却为有意义串聚类提供了方便，因为这就不需要再对整个语料进行遍历了。图 6.5 演示了有意义串 str 对应的文档链中有 N 个元素，它能够记录串 str 出现的文档以及在每个文档中出现的频次。

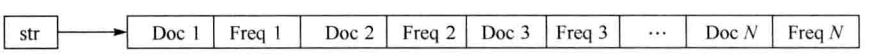


图 6.5　有意义串文档链

当需要处理大规模语料时，直接用上述方法可能会导致关系向量的维数较高，聚类的时空开销较大且向量较稀疏，进而可能影响聚类结果。所以，在聚类之前要进行特征选择，即选取出对于不同串有区分度的文档作为特征。常用的特征选择方法有计算文档频数(DF)、信息增益(IG)、期望交叉熵、互信息等。文献(Yang，1997)通过实验验证了文档频次是一种简单有效的特征选择方法。这里采用该方法对文档进行选择。类似文档频数(DF)，定义有意义串种类来表示一个文档中出现的不同有意义串的种类。

定义 6.1　有意义串种类(variety of meaningful string，VMS)　给定有意义串集合 $S=\{str_1, str_2, str_3, \cdots\}$ 和文档集合 $D=\{Doc_1, Doc_2, Doc_3, \cdots\}$，文档

Doc_i 的有意义串种类 VMS 值是指 Doc_i 中出现且包含在集合 S 中的非重复的元素的数目。

这里需要设置一个阈值 TVMS，即只保留 VMS 值大于该值的文档作为特征，进而表示有意义串与文档之间的特征关系。实验中 TVMS 取值为 5，原来的 4783 个文档经过特征选择后剩余 3846 个文档。将串和文档的关系表示成向量，向量的每一维是串在文档中出现的频次，下面是对 6 个文档 3 个串的关系向量示例。

	Doc1	Doc2	Doc3	Doc4	Doc5	Doc6
Str1	<Freq11,	Freq12,	Freq13,	Freq14,	Freq15,	Freq16>
Str2	<Freq21,	Freq22,	Freq23,	Freq24,	Freq25,	Freq26>
Str3	<Freq31,	Freq32,	Freq33,	Freq34,	Freq35,	Freq36>

上面向量中的每一维是一个整数，必须经过归一化后再进行聚类。聚类中相似度计算采用常用的余弦距离公式，聚类过程使用 ICTWMS 聚类算法包，采用 K-means 算法得到聚类结果。对两千多个串聚类产生 20 多个不同的类，除了数目较大的类和数目较小的类，有 10 个左右的类结果比较可观。下面是几个实验结果示例。

类别 1：冻土工程 高原冻土 高原病 片石 高寒缺氧 青藏铁路建设总指挥部 冻土隧道 多年冻土地段 高海拔

类别 2：环境保护工作 生态建设 环境质量 保护工作 严重破坏 构建和谐 建立和完善 扶持政策 环保工作 高污染 处理设施 环保产业 功能区 国家环保 排放总量 大气污染 化学需氧量 国家环保模范城市 环境综合整治 环境基础设施建设

类别 3：巴西队 法国队 罗纳尔多 皮尔洛 国际足联 亨利 贝克汉姆 佩雷拉 前场 格拉 卡卡 非法移民 拉姆 里皮 参加世界杯 小贝

类别 1 基本上是青藏铁路相关串，类别 2 是与环境保护相关的，类别 3 是关于世界杯足球的，从这些结果可以看出，出现一组相关的有意义串就可以确定一个话题或事件，聚类方法基本有效。

6.4 基于邻接串种类的有意义串语境聚类

新词是有意义串中的一个重要组成部分。当网络新词，如“超女”、“快男”、“郭美美”等刚出现时，人们往往不太熟悉，对新词的特性也不够了解。如果能够发现跟新词用法相似的已知词，就可以根据已知词的特点来推测这些新词的特点。如“苹果”是一种水果，出现一个新词叫“火龙果”，在我们对“火龙果”是什么不了解的情况下，如果发现“火龙果”和“苹果”的上下文语境比较相近，在出现“苹果”的地方也经常出现“火龙果”，那么说明他们的用法比较相似。由此可以推断“火龙果”是

个名词，而且可能也是个水果。所以，通过上下文分析提取有意义串的语用环境，并以此作为特征聚类，能够将语用相似的串聚在一起，这对于新词的研究是很有意义的。

语境聚类跟上面介绍的主题聚类的区别是将有意义串的上下文作为特征来完成聚类。语境聚类是在聚类过程中比较每个串的语境相似度，得到用法相似的有意义串的集合。

在第 5 章提到的重复串发现的过程中，也记录了每个串的上下文邻接对及其频次，图 6.6 是一个在 N 个邻接对中出现的串 str 的邻接链。存储时将邻接对作为一个整体来看待，但聚类时将左邻接和右邻接独立开来分析，因为邻接对的数量庞大，但是整个分布比较稀疏，噪声比较多。因此，在聚类之前将左邻接词和右邻接词提取出来，并形成两条链。

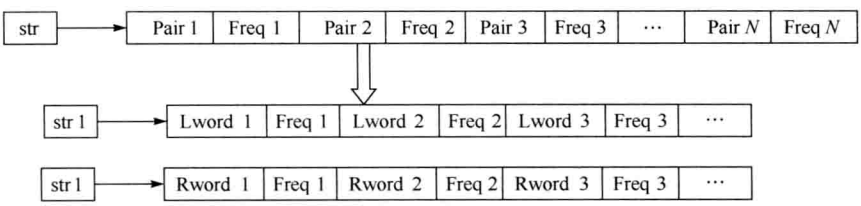


图 6.6　有意义串的邻接链

在提取算法的核心词典中，包含 6 万多个词，将这些词全部作为特征，可能会出现“高维诅咒”现象，所以要经过特征选择来降低维度。特征选择也采用类似文档频次的方法。下面给出邻接词的邻接串种类(variety of neighbor string，VNB)的定义。

给定有意义串集合 $S=\{\text{str}_1,\text{str}_2,\ \text{str}_3,\cdots\}$，左邻接词集合 $L_W=\{\text{word}_1,\text{word}_2,\text{word}_3,\cdots\}$和右邻接词集合 $R_W=\{\text{word}_1,\text{word}_2,\text{word}_3,\cdots\}$，对于每个左邻接集合 L_W 中的词，该词的 VNB 是指在整个文档集中，出现在该词右边的属于集合 S 的非重复的有意义串的数目；对于每个右邻接集合 R 中的词，该词的 VNB 是指出现在它左边的属于集合 S 的非重复串的数目。

计算每个邻接词的 VNB 后，将其按照 VNB 排序，将 VNB 值太大和太小的词去掉。在实验中，将 VNB 大于 2000 和小于 4 的邻接词过滤，最后剩余 2498 个左邻接词和 2513 个右邻接词，每个串对应一个 5011 维的邻接向量。下面是 3 个串的向量示例。

	Lword1	…	LwordN	Rword1	…	RwordM
Str1	<LFreq11	…	LFreq1N	RFreq11	…	RFreq1M >
Str2	<LFreq21	…	LFreq2N	RFreq21	…	RFreq2M >
Str3	<LFreq31	…	LFreq3N	RFreq31	…	RFreq3M >

每个串的邻接向量经过归一化计算权重，也是使用 ICTWMS 聚类算法包中的 K-means 算法完成聚类过程。聚类结果也产生了 30 多个类，下面是几个结果示例。

类别 1：扎卡维 陈水扁 梁朝伟 刘教授 男被告 佩尼亚 李宇春 刘明康 蒋先生 玲子

类别 2：副局长 相关负责人 副主席 新闻发言人 市委书记 科科长 董事局主席 钟南山 邓某 李志鹏

类别 3：价格较上月上升 同比增长 较上月上升

类别 4：国际政治奥运会期间 接受采访时 当日下午 下半场 年底前今年年初

类别 5：高度重视 宏观调控 正式宣布 监督检查 调查发现 全程监控 严重破坏 发展规划 首开纪录

从上述结果可以看出，第一个类别是人名，第二个类别是人名称谓，第四个是表示时间的短语，每个类别中的串都是在用法上有一定的相似性，对于研究新词的用法具有重要意义。

图 6.7 给出了对给定的语料库的聚类结果及每个类对应的串。

```
<?xml version="1.0" encoding="gb2312" standalone="yes" ?>
- <LJCluster-Result>
    <clusnum>500</clusnum>
  - <clus id="0">
      <feature>奥巴马 攻击型潜艇 胡德堡 理财师 杨洁篪 詹姆斯 丹尼斯 汉森 京都议定书</feature>
    - <docs num="90">
        <doc>..\TestCorpus\笑称若滑倒是国际事件.txt</doc>
        <doc>..\TestCorpus\外资行理财师流动秘密：完不成指标三个月.txt</doc>
        <doc>..\TestCorpus\气候教父断言哥本哈根大会注定失败.txt</doc>
        <doc>..\TestCorpus\美出台阿富汗新战略 增兵旨在尽快摆脱战争泥潭.txt</doc>
        <doc>..\TestCorpus\美军基地枪击案嫌犯开枪时高喊“真主伟大”.txt</doc>
        <doc>..\TestCorpus\美驻华使馆公布奥巴马中文译名改称欧巴马.txt</doc>
        <doc>..\TestCorpus\美增兵阿富汗被批 奥巴马政府要员急辩护.txt</doc>
        <doc>..\TestCorpus\马克回忆与奥巴马哥哥见面 称哥哥是家中核心.txt</doc>
        <doc>..\TestCorpus\美印关系中的中国因素.txt</doc>
        <doc>..\TestCorpus\羽泉谈为奥巴马献唱 秘密排练奇装异服不准上身.txt</doc>
        <doc>..\TestCorpus\媒体称奥巴马向阿增兵讲话将提及撤军时间表.txt</doc>
        <doc>..\TestCorpus\杨洁篪外长谈胡锦涛主席出席四大峰会成果.txt</doc>
        <doc>..\TestCorpus\美国中情局获海外间谍指派权(图).txt</doc>
        <doc>..\TestCorpus\美国就业峰会难解失业困局.txt</doc>
        <doc>..\TestCorpus\媒体称奥巴马阿富汗新战略面临左右为难困境.txt</doc>
        <doc>..\TestCorpus\三四批编队共同护航.txt</doc>
        <doc>..\TestCorpus\美国媒体偏爱奥巴马超布什 麦凯恩阵营大.txt</doc>
        <doc>..\TestCorpus\美国是否会将我列为汇率操纵国？.txt</doc>
        <doc>..\TestCorpus\美国时代周刊评最着名道歉 伍兹居首奥巴马居次.txt</doc>
        <doc>..\TestCorpus\美报告要求奥巴马施压中国减少对台导弹部署.txt</doc>
        <doc>..\TestCorpus\事先练10天.txt</doc>
        <doc>..\TestCorpus\美民众示威抗议增兵阿富汗 大呼让士兵回家团聚.txt</doc>
        <doc>..\TestCorpus\外交部就遣返赖昌星及中国货轮遭劫持等答问.txt</doc>
        <doc>..\TestCorpus\中国企业家提问奥巴马.txt</doc>
```

图 6.7 聚类结果示例

6.5　有意义串对分类的改进

现有的文本表示方法多使用词语或 N-gram 作为特征，这两表示方式都有各自的缺点——词语表示的缺点是原始文本中的大量信息被丢失(如段落、句子、词序等信息都将被忽略)、N-gram 表示的缺点是它的计算量大、数据噪声大、存在过学习等。

用有意义串作为特征能改进分类效果。首先，因为有意义串可以发现一些跟类别相关的新用语、人名等特征词汇。例如，在科技文章中，“神舟六号”、“杨利伟”等词可以标示天文科学类文档，“非典”可以标示生命科学类文档，“全球变暖”标示自然科学类文档。另外，与单个的词相比，有意义串包含更多的信息量，能够更准确地表达语义概念，减少歧义产生的情况。例如，“机器”和“学习”两个词的类别信息不是很明显，而“机器学习”则很明显是学术领域的词汇。又如，“移动通信”比“移动”来说更具体，看到“移动通信”就知道说的是运营商“移动”，而不是动词的“移动”。另外，有意义串作除了能作为特征改进分类效果以外，其属性值也可以用来标示特征的权重。这里选择有意义串邻接类别值和词长度这两个属性值来计算特征权重。邻接类别值能够反映一个字符串的语用丰富程度，语用环境越多样，说明该字符串的重要程度越高。词长度越大的字符串语义更明确，具有更高的分类能力。因此，选择这两个属性值来改进特征权重的计算方式。

一般地，分类可分为训练阶段和分类阶段。在训练阶段，一组已被标记好的文档集合被用来训练分类器。在分类阶段，分类系统用已经训练好的分类器将给定的文档标记为一个预先设定好的类别。加入有意义串的分类器训练过程如图 6.8

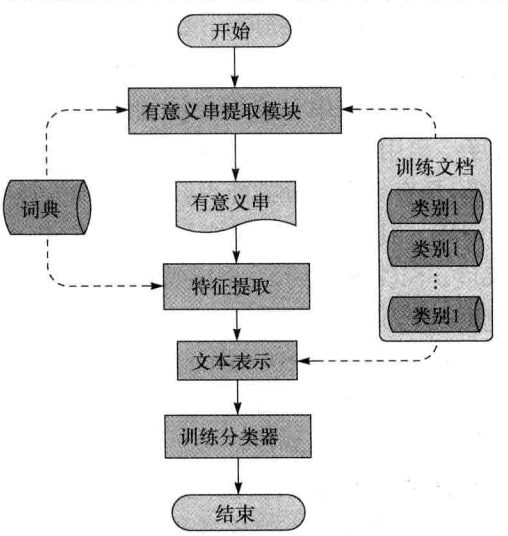


图 6.8　训练分类器流程图

所示。首先，利用有意义串提取模块，从训练文档集合的所有文件中提取有意义串并生成有意义串集合 MSSet。接着，将 MSSet 和词典 D 中的词条输入到特征提取模块中，利用特征提取方法从 MSSet 和 D 中提取有效的特征向量 $\overline{T}=\{t_1, t_2, \cdots, t_m\}$。利用有意义串的属性值对特征向量 $\overline{T}$ 中的每个特征赋权值并将每个文档表示成 m 维的向量 $\overline{d}_j=\{w_{j,1}, w_{j,2}, \cdots, w_{j,m}\}$，最后用生成的文档向量集合训练分类器。生成的分类器即可对新的文档进行分类。

分类过程如图 6.9 所示。这里同样利用有意义串的属性值，对特征向量 $\overline{T}$ 中的每个特征赋权值，并将每个待分类的文档被表示成描述向量 $\overline{X}_K=\{a_{k,1}, a_{k,2}, a_{k,m}\}$。把文档向量 $\overline{X}_k$ 输入到训练好的分类器中，分类器为该文档称为每种既定类别的可能性打分，将该文档标注为分数最高的类别。

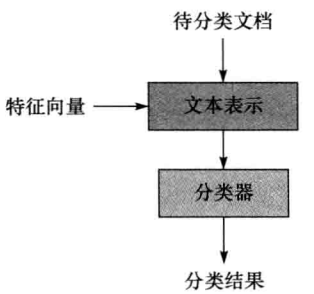


图 6.9 分类预测结果流程图

在有意义串对分类系统的改进中，有意义串属性对特征权值的改进是其中的核心。我们选择有意义串邻接类别值和词长度这两个属性值来计算特征权重。

字符串 S 的邻接类别 AV 值是 S 左边邻接词的类别数目和右边邻接词的类别数目的最小值，而词长度是字符串包含词典词的个数。邻接类别值能反映一个字符串的语用丰富程度。语用环境越多样，说明该字符串的重要程度越高；词长度大的字符串语义更明确，具有更高的分类能力，因此要加大该字符串的权重。

假设 $S=W_1W_2\cdots W_n$，其中 W_i 为词典词，AV(S)表示 S 的邻接类别值，Len(S)表示 S 的词长度。SIG(S)表示一个字符串的重要性，这里用 S 的 AV 值和词长度来衡量它的重要性，则 SIG(S)为

$$\mathrm{SIG}(S)=\mathrm{AV}(S)*(\log \mathrm{Len}(S)+1)$$

字符串 S 可以选择只用 SIG(S)作为特征权重，也可以用 SIG(S)与传统的 TF 和 IDF 相结合来表示特征的权重。

为验证其对分类系统的性能改进，我们进行了实验，采用的语料为 TanCorp 中文分类语料，包含 12 大类、60 小类。本实验采用的是 TanCorp 的电脑和科技类语料。电脑类语料包含“电脑软件”、“电子商务”等 6 个子类，2943 个文档，大小为

4.17M。科技类语料包含“自然科学”、“天文科学”等 4 个子类，1040 个文档，大小为 1.97M。用这些语料验证有意义串对分类的改进效果。除此之外，实验还采用了四种类别的专利语料，分别为作业/运输、化工、物理、机械/照明。用这些语料验证训练语料的规模对分类效果的影响。实验中均采用 70%的文档作为训练文档，30%的文档作为测试文档。实验主要分为三部分：首先，将有意义串作为特征，验证分类的改进效果，主要比较只用词典词作为特征和加入有意义串作为特征的分类准确率；第二，由于语料的规模对有意义串的提取效果有影响，进而对分类有影响，因此测试语料规模对分类的影响。最后用有意义串的属性修正特征权重，验证分类的改进效果。比较 TFIDF 权值赋值方式和加入有意义串属性的赋值方式下分类准确率的变化。这里用分类的准确率来衡量分类效果。分类正确的文档数目越多，说明分类的效果越好。假设有 K 个类别，A_k 表示被分类器判定为类别 k 且实际类别为 k 的文档数目，B_k 表示为被分类器判定为类别 k，但实际类别不为 k 的文档数目，则准确率 Precision 的定义为

$$\text{Precision} = \frac{\sum_{k=1}^{K} A_k}{\sum_{k=1}^{K} A_k + \sum_{k=1}^{K} B_k} \times 100\%$$

实验将只用词典词作为特征的分类算法作为基线，用加入有意义串作为特征后分类准确率的提高来证明有意义串对分类的改进。实验分别采用 NaiveByes、LibSVM 和 C4.5 作为分类器，比较加入有意义串对不同分类器的影响。表 6.3 是电脑语料的实验结果。表 6.4 是科技语料的实验结果。

表 6.3　电脑语料加入有意义串的分类准确率

	Dict	Dict+MS
NaiveBayes	68.9433	71.8315(+0.042)
LibSVM	73.6323	74.21(+0.008)
C4.5	63.8464	64.3221(+0.007)

表 6.4　科技语料加入有意义串的分类准确率

	Dict	Dict+MS+IG
NaiveBayes	79.2916	79.7033(+0.005)
LibSVM	73.4332	75.6131(+0.030)
C4.5	72.2071	71.7984(−0.007)

可以看出，对于 NaiveBayes 分类器和 LibSVM 分类器，向词典中加入有意义串有利于提高分类的准确。其中，NaiveBayes 分类器对电脑语料分类的改进效果

为 4.2%。电脑领域的语料分类改进效果要明显一些，主要的原因是电脑语料中存在被普遍使用的公司名和机构名，如“趋势科技”、“运营商”等，还有一些网络流行的新词“玩家”等。这些词汇能够有效的提高分类效果。

表 6.5 是作为分类特征的有意义串。这些词包括人名(杨利伟)、地名(苏门答腊)、机构名(微软、信产部)以及新出现的流行语(非典、玩家)等。这些词能明显的表示类别，对分类有较好的改进作用。可见，用有意义串作为特征将有利于提高分类器的分类效果。

表 6.5 作为分类特征的有意义串

语料	作为分类特征的有意义串
电脑语料	趋势科技、赛门铁克、特洛伊木马(电脑病毒) 运营商、信产部、发改委、微软(电脑科技) 网络游戏、玩家(电脑游戏) 数码相机、像素、手机牌照(电子商务)
科技语料	神舟六号、杨利伟、探月(考古科学) 苏门答腊、高程测量(自然科学) 非典、流感病毒(生命科学)

另外，由于语料的规模对有意义串的提取效果有影响，语料越大有意义串的提取效果越好，有意义串的稀疏性也会有所改善。实验采用专利申请的四个类别的语料，分别为作业/运输、化工、物理、机械/照明。从这四个语料取不同数量的文档做实验，采用 NaiveBayes 作为分类器。表 6.6 中第一列数据是从每个类别的语料中取文档的数目。从实验结果中可以看出，随着语料规模的扩大，有意义串对分类的改进效果不断提高。这主要是由于语料规模较小时，找出的有意义串的分布比较稀疏。随着语料规模的增加，可以找到更多、更有效的有意义串作为分类特征。另外，随着规模的增加，有意义串在更多的文档中出现，增强了有意义串的影响力。实验说明在加入有意义串的分类系统中，训练文档应达到一定的数量，以更好地提高分类效率。

表 6.6 语料规模对改进效果影响

规模	Dict	Dict+MS
500	78.85	**79(+0.002)**
1000	80.675	**81.275(+0.007)**
1500	80.9667	**81.5667(+0.007)**
2000	80.85	**81.35(+0.006)**

最后，再来看看有意义串属性对特征权值的改进。我们用 NaiveByes 作为分

类器，分别用 TF* IDF、SIG、SIG* TF* IDF 作为分类特征的权值，对电脑语料和科技语料进行分类。比较三种不同权重赋值方式得到的分类准确率。从表 6.7 中可以看出，用 SIG 作为权重的效果要好一些。电脑语料的分类效果提高 1.8%，科技语料的效果提高 4.1%，但用 SIG* TF* IDF 作为权值并没有比只用 SIG 作为权重的效果好。总的说来，使用 SIG 作为权重对分类是有改进作用的，说明有意义串的属性有利于提高分类效果。与不使用有意义串的分类方法相比较，加入有意义串使电脑语料的分类准确率提高了 3.7%，科技语料的分类准确率提高了 5.9%。该结果说明将有意义串应用于分类中能有效地提高分类准确率。

表 6.7　有意义串属性对特征权重的改进

语料	TF* IDF	SIG	SIG* TF* IDF
电脑语料	70.2345	**71.4917(+0.018)**	70.6082
科技语料	80.6731	**83.9423(+0.041)**	79.1346

6.6　本章小结

本章将关键词提取技术与检索结果的特点结合，提出将词频、词性和互信息等特征相结合的多特征融合评价方法抽取关键词，并以关键词作为基础特征进行层次化结果聚类。本章给出的方法属于基于标签的聚类，采用关键词提取的方法进行标签选择，解决传统方法获取类名可读性差的问题。以关键词作为基础特征进行聚类以及类别合并，获得可读性强的层次化的聚类结果。同时，也给出了基于有意义串挖掘的文本聚类和分类算法研究。

相关工作与扩展阅读

目前已有各种各样的聚类方法，按照聚类方法的特点可以分为基于文档内容的聚类与基于标签的聚类(Toda et al.，2005)。

首先是基于文档内容的聚类。该方法主要采用传统的聚类方式，使用搜索引擎返回的标题、摘要替代全文进行聚类进行效率优化。SCATTER/GATHER (Marti et al.，1996)是第一个 Web 检索结果聚类系统，使用夹角余弦定理计算两个文档的相似性，采用 Fractionation 算法进行聚类。WebCat(Giannoni et al.，2003)使用传统的K-means 算法获得聚类结果。由于标题与摘要信息较少，数据稀疏问题严重，导致聚类结果较差，分析得到的聚类名称不易理解。目前文本聚类方法已经较为成熟，但是搜索结果聚类不同于普通文档的聚类，不能直接套用传统的聚类方法。首先，普通文档集合中的两篇文档没有必然的联系，而搜索结果集合中的文档主题都是围绕关键词的，文档之间都有一定的关联；其次，由于效率的原因，

搜索结果聚类使用的是文章的标题与摘要，包含的信息量少，篇幅短小；最后，部分网站采用搜索引擎优化技术，导致某些垃圾串频率较高。

其次是基于标签的聚类。首先，从文档集中抽取有价值的短语、词组和片段作为标签作为类别名，然后将文档分配给这些类别。SNAKET（Ferragina et al.，2005）利用知识库对词语进行评分和选择，选出权重高的词语形成基类，并利用权重较低的词语指导进一步的聚类。词汇相关度聚类利用相关度高的词对指导聚类。SHOC（Zhang et al.，2004）通过后缀词组得到连续的词语作为标签，并利用SVD获得层次的类结构。张刚等（2008）采用图的方式，统计文档共现词频，选择词频高的词作为基类，然后通过图的连通关系进行进一步的聚类。蒋昌金等（2010）提出了一种基于组合词和同义词集的关键词提取算法。该算法首先利用组合词识别算法极大地改进分词效果，能识别网页上绝大多数的新词、未登录词，为提高关键词自动抽取准确率奠定了坚实的基础；同时利用构造的同义词集，合并同义词的词频，避免了同义词在输出结果中同现。罗准辰等（2009）提出将关键词提取分为两个问题进行处理，即关键单词提取和关键词串提取，设计了一种基于分离模型的中文关键词提取算法。赵小谦等（2012）提出了通过构建概念书的方法以计算相似度的方法。翟延冬等（2012）提出了一种短文本语义相似性计算算法，算法使用 wordNet 和 Brown 文集来计算文本中的概念相似性，在此基础上提出了一个新的方法综合考虑概念和句法等信息来计算短文本的语义相似性。

参 考 文 献

黄玉兰，龚才春，许洪波，等．2008．基于局部性原理的有意义串提取方法// 第四届全国信息检索与内容安全学术会议论文集．

蒋昌金，彭宏，陈建超，等．2010．基于组合词和同义词集的关键词提取算法．计算机应用研究．27(8)：2853-2856．

罗准辰，王挺．2009．基于分离模型的中文关键词提取算法研究．中文信息学报，23(1)：63-70．

翟延冬，王康平，张东娜，等．2012．一种基于 WordNet 的短文本语义相似性算法．电子学报，40(3)：617-620．

张刚，刘悦，郭嘉峰，等．2008．一种层次化的检索结果聚类方法．计算机研究与发展，45(3)：542-547．

赵小谦，郑彦，储海庆．2012．概念树在短文本语义相似度上的应用．计算机技术与发展，22(6)：159-162．

Ferragina P，Gulli A. 2005. A personalized search engine based on web-snippet hierarchical clustering// Proceedings of Special Interest Tracks and Posters of the 14th International Conference on World Wide Web.

Giannoni F，Nanni M，Peclreschi D. 2003. Webcat：automatic categorization of web search re-

sults// Proceedings of the SEBD03.

Marti A H, Jan O P. 1996. Reexamining the cluster hypothesis: scatter/gather on retrieval results// Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.

Toda H, Ryoji K. 2005. A search result clustering method using informatively named entities// Proceedings of the 7th annual ACM international workshop on Web Information and Data Management.

Yang Y M, Jan O P. 1997. A comparative study on feature selection in text categorization// Proceedings of 14th International Conference on Machine Learning.

Zhang D, Dong Y. 2004. Semantic, hierarchical, online clustering of web search results// Proceedings of the 6th Asia Pacific Web Conference (APWEB).

Zhao J, He J. 2006. Learning to generate labels for organizing search results from a domain specified corpus// Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence.

第 7 章　大数据文本自动摘要

7.1　概　　述

目前从大数据中获取信息的方式主要有数据库查询和文本搜索等方法。数据库主要用于处理结构化数据；信息检索主要用于对非结构化信息的处理，信息检索返回的结果集的数量一般非常巨大，用户要从检索结果中找到有用的信息仍然十分困难，因此对检索结果自动提取摘要就显得尤为重要。本章实现基于关键词的摘要提取和面向主题的摘要系统，并在提高摘要的处理效率、基于关键词提取摘要、摘要中句子间的冗余去除方面进行了研究。首先，在提高效率方面，引入搜索引擎的倒排链表结构统计词、句子的特征，并使用双数组 Trie 树存储分词词典和用户主题词表，以期提高词的查找效率；将关键词提取与摘要提取相结合，在关键词提取中引入词邻接类别、词的位置局部性，以便提高高频词和低频词的质量。其次，在去除句子冗余度方面，提出句子之间的包含度概念，能够对文章中存在包含关系的一类句子进行有效的排重，降低文摘的冗余度，提高文摘质量。

7.2　相关工作综述

信息检索研究源于图书馆的资料查询和文摘索引工作。随着计算机网络技术的迅速发展和普及，信息检索研究的内容已经从传统的文本检索扩展到包含图片、音频、视频等多媒体信息的检索。检索对象从相对封闭、稳定一致，由独立数据库集中管理的信息内容扩展到开放、动态，更新速度快、分布广泛、管理松散的网络内容(高凯，2010)。从 20 世纪 60 年代开始，许多研究者提出了不同的检索模型，其中比较有代表性的模型有布尔模型、向量空间模型、概率检索模型等，并产生了 SMART、OKAPI 等著名的原型系统。这些检索模型和原型系统有力地推动了信息检索的研究和发展。

自动文摘是利用计算机实现文本预处理、语义分析和摘要自动提取的技术。Luhn(1958)揭开了计算机实现自动文摘研究的序幕。因受计算机硬件条件和应用需求的限制，最初自动文摘技术并没有得到足够重视。1995 年国际著名学术刊物 Information Processing & Management 发表的论文(Jones，1995)标志着自动文摘时代的到来，自动文摘技术研究由此进入了新的发展时期。有的学者(Sparck，2007)把文摘过程划分成对源文本进行处理从而获取源文本表达，对源

文本的表达进行变换，从而获取文摘表达和文摘生成的三个阶段：第一阶段是对源文本进行分析处理，将源文本切分成获得文摘的基本片段，如句子、段落等，以及影响这些基本片段的其他片段，并统计相应信息；第二阶段对第一阶段获得的基本片段进行计算；第三阶段根据前面获得的片段的相关信息选取文摘片段并按一定规则组合生成文摘。有的学者将自动文摘划分为不同的类型(Zhang et al.，2008)，根据所采用的路线不同，自动文摘方法可分为基于统计的方法和基于机器学习的方法两大类，下面分别予以概述。

在基于统计的方法中，一类是基于词项的统计方法，这种方法直接采用词项特征进行句子或概念的度量与识别，利用句子包含特征的权重度量句子重要度，按重要度给句子排序，再根据排序从文摘中进行文摘句的选取，最后按照在文档中出现的顺序输出文摘。这种方法的基本出发点是，重要的句子是由重要的词组成，而重要的句子更适合被选择为文摘句。基本的统计方法很少考虑句子之间的关系，而是主要关注利用词频信息来反映内容的重要性，其中以质心法(Centriod)(Radev et al.，2004)、最大边缘相似度方法(MMR)(Carbonell，1998)为代表。周进华等提出将词共现图方法用于多文档自动摘要(周进华等，2009)。第二种方法是在进一步发现与挖掘特征的基础上，用更加丰富的统计特征进行文档中重要句子或概念的识别，从中抽取出重要内容单元形成文摘。一方面改进当前向量空间模型单纯利用词项的问题，引入 N 元组等来决定句子的统计特性；另一方面采用基于统计相关性方法来识别话题标志特征等信息。同时，更多语言学特征需要考虑进来，它们用来更好地度量句子的重要度，其中以 Lexical Chain(Barzilay et al.，1997)等方法为代表。第三种方法是基于文档结构关系的多文档自动文摘方法，这种方法充分利用文本的结构信息从文档中提取文档的重要概念(句子、子话题、类中心)，是当前研究的热点。Erkan 等(2004)提出了 LexRank 方法，采用社会网络中心度的思想来度量文档集中的重要概念，将 PageRank 算法的思想引入自动文摘中，在定义的图中，每个点代表一个句子，如果两个句子的相似度大于某一临界值，则有一条边连接这两个句子。每个句子的重要性由与它相邻的所有句子的重要性决定，即一个句子如果与很多重要度很高的句子相邻，则它本身的重要度也很高。

在基于机器学习(machine learning，ML)的多文档自动文摘方法中，可以采用隐性语义分析(latent semantic analysis，LSA)和支持向量机(support vector machine，SVM)等方法，通过监督和非监督的方式来生成多文档文摘。通常对于给定的文本特征，可以采用当前的各种机器学习的方法在特征的选择和组合的基础上找到一种适合文摘的方法。

根据输入文档的数量划分，自动文摘又可分为单文档摘要和多文档摘要。根据文摘和原文的关系划分，可划分为摘录型文摘(extract)和理解型文摘(abstract)，前者从原文中抽取片段形成文摘，而后者则是对原文主要内容重新组织后

形成的。根据文摘的应用划分，可将自动文摘分为通用型文摘(generic)和面向用户查询的文摘(query-oriented summarization)，前者客观反映文档的主要内容，而后者则着重于用户感兴趣的内容。上述各类型的划分不是互斥的，如摘录型文摘既可以在单文档中实现，也可以在多文档中实现。20 世纪 90 年代中期以前，自动文摘领域的主要研究内容是单文档文摘技术，这个时期主要的方法是文本抽取和信息抽取(Sparck，2007)。随着互联网技术的迅猛发展，跨文本信息融合的需求也越来越强烈，20 世纪 90 年代中期正式开始了通用领域的多文档文摘的研究，到目前为止在英文多文档自动文摘研究上已取了较大成绩，比较优秀的成果有 Lexical Chain(Barzilay et al.，1997)方法、MMR(Carbonell et al.，1998)方法和质心法(Radev et al.，2004)。国内对多文档文摘的研究起步较晚，尤其是中文自动文摘的研究，受大规模测试数据集和评价工具等的限制，目前的研究还处于起步阶段，有待进一步深入。刘挺等曾将自动文摘方法概括为自动摘录、基于理解的自动文摘、信息抽取和基于结构的自动文摘。在这四种方法中，自动摘录是其他算法的基础，也是最简单、最易于实现的方法。这里不采用如此详细的划分，只是简单划分为基于抽取的自动文摘和基于理解的自动文摘两类方法。

7.2.1 基于抽取的自动文摘

目前自动文摘方法的研究大多都是基于抽取的文摘。基于抽取的自动文摘将文本视为句子(段落或者其他文本片段，下面仅以句子代替)的线性序列，将句子视为词的线性序列。它通常分 3 步进行。

(1) 对源文本按一定规则进行切分，获取源文本的句子序列、句子的词序列表达。

(2) 计算词、句子的权重，对原文中的所有句子按权值高低降序排列，权值最高的若干句子被确定为文摘句。

(3) 将所有文摘句按照它们在原文中的出现顺序输出。抽取文摘的关键问题是句子的选择及句子的排序问题。

句子的选择方法有很多，但选取的原则是一致的：尽量用最重要的句子来反映文档集的话题，即话题相关度要尽量高，同时抽取出来的文摘句之间内容冗余度应尽量小。所以句子选择又分为句子权重计算和句子相似度计算两个问题。单文档文摘的句子一般都按照句子在原文中出现的顺序排列，而在多文档文摘中，大部分采用时间顺序排列句子(Zhang et al.，2008)。

1. 句子权重计算

句子是否选择为文摘句的根据往往是句子的权重。计算句子权重的因子有句子所包含词的权重、句子所在段落的权重、句子在段落中的位置、句子与文档中其

他句子的相似程度等。计算句子权重、词的权重、段落权重是一个交叉进行、相互影响的过程。从源文本的角度看，计算句子权重主要依据如下特征（刘挺等，1999），即词频、词项在文档中的分布规律、标题、位置、句法结构、线索词、指示性短语等。其中词频、词项在文档中的分布规律、标题主要通过影响词项权重影响句子权重，位置、句法结构、线索词、指示性短语直接作用于句子权重。

（1）词频。Luhn 在开创自动文摘领域时提出采用词频统计来摘要的思想，但仅靠词频来标示词项的权重还不够。1995 年美国 GE 研究与开发中心的 Lisa 等完成了 ANES(automatic news extraction system)系统，该系统采用相对词频作为词的权值。

（2）词项在文档中的分布规律。词项在文档的大部分段落中均匀分布还是在某几个段落中集中出现对词揭示主题的作用有相当大的影响。

（3）标题。标题中的关键词对揭示文章主题有重要作用，因此出现在标题中的关键词权重应增加。

（4）位置。调查表明，段落的论题是段落首句的概率为 85%，是段落末句的概率为 7%。因此，有必要提高处于特殊位置的句子的权值。

（5）句法结构。句式与句子的重要性之间存在着某种联系，例如文摘中的句子大多是陈述句，而疑问句、感叹句等则不宜进入文摘。

（6）线索词。Edmundson 的文摘系统有一个预先编制的线索词词典，词典的线索词分为取正值的褒义词（bonus words）、取负值的贬义词（stigma words）、取零值的无效词（null words）。句子的权值等于句中每个线索词的权值之和。

（7）指示性短语。英国 Lancaster 大学的 Paice 提出根据各种“指示性短语”选择文摘句的方法。与线索词相比，指示性短语的可靠性要强得多。

2. 句子相似度计算

目前常用的句子相似度计算模型主要有向量空间模型、查询似然模型和翻译模型。向量空间模型因为其计算简单而被广泛应用。向量空间模型将句子相似度定义为两个向量间夹角的余弦值，即

$$\mathrm{sim}(d,q)=\frac{d\cdot q}{|d|\times|q|}=\frac{\sum_{i=1}^{t}w_{i,d}\times w_{i,q}}{\sum_{i=1}^{t}w_{i,d}^{2}\times\sqrt{\sum_{i=1}^{t}w_{i,q}^{2}}}$$

其中，$|d|$ 和 $|q|$ 分别代表文档向量与话题向量的模；$w_{i,d}$ 和 $w_{i,q}$ 分别代表文档向量与话题向量中的第 i 个词的权重。

3. 句子先后排序

为了确保文摘句子的一致性和连贯性，需要排列文摘句的先后顺序。目前采用的排序方法通常有两种。一种是时间排序法，一般选定某一个时间为参考点，然后计算其他相对时间的绝对时间，如可以使用出版日期作为参考点，并为本周内的日期、以往或今后周内的日期和“今天、昨天、昨晚……”等表达方式计算绝对时间。另一种是扩张排序法，其目的是试图通过将有一定内容相关性的主题放在一起，以此来降低摘要的不流畅性。抽取文摘只对有用的文本片段进行有限深度的分析，其效率和灵活性较高，因而适合网络信息等大规模文本，仅关注文档主题而对文摘连贯性要求不高的场合。目前抽取方法得到了广泛的应用，出现了许多有效的方法，如搜索引擎领域的摘要生成。总体来说，抽取方法得到的文摘效果仍不尽如人意，效率高的算法文摘质量较差，而质量较好的算法又因为其计算量巨大很难应用于各种场合。

7.2.2 基于理解的自动文摘

基于理解的自动文摘与抽取文摘的主要区别是文摘内容是否出自原文、对文档语义分析的深度不同。基于理解的文摘内容不完全出自原文，它利用语言学知识获取语言结构、领域知识进行判断和推理，得到文摘的意义表示，最后从意义表示中生成文摘。基于理解的自动文摘结合已有知识库对文档的语法、语义等进行深入的分析挖掘，以期“理解”原文；基于抽取的自动文摘仅对文档的词汇等进行浅层次的分析。理解文摘的不足在于领域严格受限。造成领域受限的原因如下。

(1) 面向大规模真实语料的语法语义分析技术尚未完全成熟，因此如果想获得高质量的语言分析结果，就必须将待处理的语料限制在某个范围之内。

(2) 理解文摘方法的基础是框架等知识表示，框架需要根据领域知识预先拟定，因此如果想把适用于某个领域的理解文摘系统推广到另一领域，则需重新拟定框架，这种填充和组织领域知识的沉重负担使理解文摘难以移植。

7.3 基于关键词提取的自动摘要

基于关键词提取的自动摘要整体流程如图 7.1 所示。

7.3.1 文本预处理

网页文本的内容一般是通过一定的方法对网页进行正文提取得到的，因而获得的正文难免会有许多不规范、无意义的字符或字符串(如 等)，甚至会有一些未处理好的 html 标签；有些网页本身的内容格式非常复杂，例如图 7.2 中的

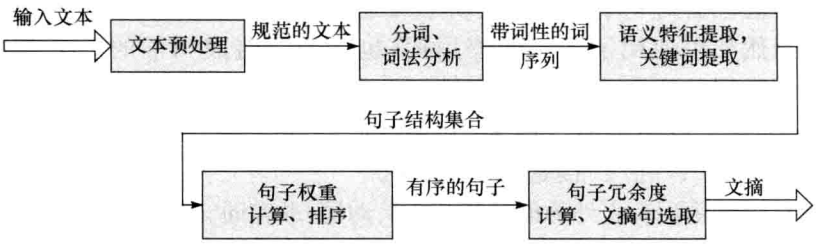
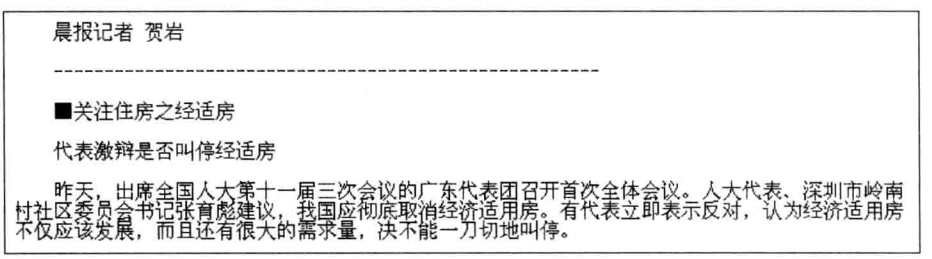


图 7.1　基于关键词的自动摘要流程

“----------”与“■”不但对文摘生成没有意义，而且还可能导致分词时因为字典中没有该词导致按字节切分，出现大量的无意义字符，增加摘要词频、词性统计等后面环境的处理时间。在预处理阶段，需要去除这些无意义的字符串。另一方面，文本中出现的数字有全角、半角的区别，直接分词会导致相同的数字因为全角、半角的不同而产生不同的词，而这对文摘提取来说也会产生影响，其他一些字符也有全角、半角的区别，英文字母有大小写的区别等；在预处理阶段，需要将这些字符转化为统一的格式，然后再做下一步的处理。这阶段将这些特殊字符分为两类：一类是无意义串，另一类是特殊格式串。对于无意义串，直接将其从原文中删除，而对于特殊格式串，则将其转化为文摘系统中统一使用的标准格式。


晨报记者 贺岩

■关注住房之经适房

代表激辩是否叫停经适房

昨天，出席全国人大第十一届三次会议的广东代表团召开首次全体会议。人大代表、深圳市岭南村社区委员会书记张育彪建议，我国应彻底取消经济适用房。有代表立即表示反对，认为经济适用房不仅应该发展，而且还有很大的需求量，决不能一刀切地叫停。

图 7.2　正文中存在部分噪声信息图示

7.3.2　停用词处理

词语作为表达句子或文档意义的基本单位，能够表达不同的句子、文档。但是，不同词语的表达能力差异很大，在文摘生成和关键词提取过程中，过滤掉停用词可以提高系统的计算效率和文摘的质量。停用词的去除相对来说比较简单，主要方法如下。

(1) 人工整理一份类似于黑名单的停用词表。文档中如果出现停用词表中的词，则将它删除。某个词成为停用词主要依靠以下两条规则：一是文本中出现频率很高，但是分布十分广泛、意义十分“一般”的词。例如，“我”、“就”之类词几乎在每个文档上均会出现，这样的词对于区分句子不同的重要程度

意义不大，甚至会对句子表示的真实意义造成干扰。因此，会忽略这类特定的常用词，当然如果使用了太多的停用词，也可能会造成句子的意义表达不准确，因此这类词的添加需要非常谨慎。二是频率很高，但实际意义又不大的词。这一类主要包括了语气助词、介词、副词、连词等，通常自身并无明确意义，如“的”、“在”、“和”、“接着”之类的词。

(2) 对于停用词表之外的词，保留名词、动词、形容词、副词。

(3) 在文档标题或者主题描述中出现的不在停用词表中的词予以保留，除此之外的所有词语均予以过滤。

停用词表的存储及过滤处理将在下节所述的双数组 Trie 树中介绍。

7.3.3 双数组 Trie 树

Trie 树又称键树、数字查找树，是一种树形结构，用于保存大量的字符串。Trie 树是中文匹配分词算法中词典的一种常见实现，其本质上是一个确定的有限状态自动机，每个节点代表自动机的一个状态。在词典中这些状态包括“词前缀”、“已成词”等，其优点是利用字符串的公共前缀来节约存储空间，能最大限度地减少无谓的字符串比较，查询效率比哈希表高。Trie 树具有如下三个基本特性。

(1) 根节点不包含字符，除根节点外每一个节点都只包含一个字符。

(2) 从根节点到某一节点，路径上经过的字符连接起来，是该节点对应的字符串。

(3) 每个节点的所有子节点包含的字符都不相同。

图 7.3 是对五个单词“say she shr he her”构造的 trie 树。

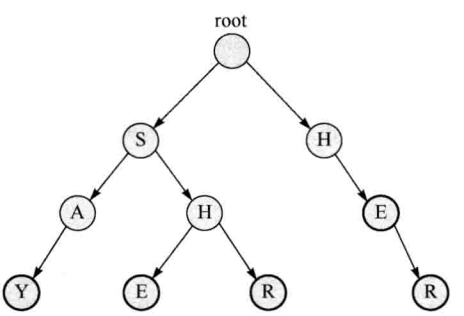


图 7.3 单词构造的 Trie 树

Trie 树中的查找分为两种：一种是查找一个词在 Trie 树中是否存在，另一种是在一个字符串中查找能匹配 Trie 树中模式串的子串。第一种用在停用词过滤的处理中，第二种主要用在分词中。在 Trie 树中查找一个词的一般过程是从根结点出发，沿和词语中的字符逐层向下，直至词语的最后一个字符或叶子结点，若未到达叶子结点且该节点是终结点，则查找成功；若到达叶结点且叶子结点中的关键

字和词语的最后一个字符相匹配，则查找成功；若词语中某个字符在树中找不到相应的结点，或者已查找到 Trie 的叶子结点，但是词语还未匹配到最后一个字符，则查找失败。第二种匹配与第一种的区别仅在于查找到 Trie 树叶子结点的处理上，如果查找到叶子结点且字符串未匹配到结尾，则在串中记录下这个位置，然后指针往后移，并从 Trie 树的根结点开始查找。

相对而言，Trie 树是一种简单高效的数据结构，理解起来比较简单，但它的内存消耗非常大。为了减少 Trie 树结构的空间浪费，同时保证 Trie 树查询的效率，有研究者提出了用三个线性数组表示 Trie 树的方法，并在此基础上进一步改进，用两个数组来表示 Trie 树，也就是双数组 Trie 树，如图 3.5 所示。

双数组 Trie 是 trie 树的一个简单而有效的实现，由两个整数数组构成，一个是 base[]，另一个是 check[]，这两个数组是一一对应的关系。设数组下标为 i，如果 base[i]、check[i]均为 0，表示该位置为空。如果 base[i]为负值，表示该状态为词语。Check[i]表示该状态的前一状态，t = base[i] + a，check[t] = i。双数组 Trie 树解决了 Trie 树消耗空间极大的缺点，当然也带来了比较漫长的构建和更新时间。在诸如分词等领域，模式串的变化周期很长，而数据规模很大，则双数组 Trie 树无疑是最好的选择。下面举例来说明用双数组 Trie 构造分词算法词典的过程。假定词表中只有“啊”、“阿根廷”、“阿胶”、“阿拉伯”、“阿拉伯人”、“埃及”这几个词，如表 7.1 所示。

表 7.1　双数组 Trie 树举例

下标	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Base	−1	4	4	0	0	0	0	4	−9	4	−11	−12	−4	−14
Check	0	0	0	0	0	0	0	2	2	2	3	8	10	13
词缀	啊	阿	埃					阿根	阿胶	阿拉	埃及	阿根廷	阿拉伯	阿拉伯人

用上述方法生成的双数组，将“啊”、“阿”、“埃”、“阿根”、“阿拉”、“阿胶”、“埃及”、“阿拉伯”、“阿拉伯人”、“阿根廷”均视为状态。每个状态均对应于数组的一个下标。例如，设“阿根”的下标为 i = 8，那么 check[i]的内容是“阿”的下标，而 base[i]是“阿根廷”的下标的基值。“廷”的序列码为 x = 8，那么“阿根廷”的下标为 base[i] + x = base[8] + 8 = 12。

除在分词中使用双数组 Trie 树结构存储词典外，停用词表的管理使用双数组 Trie 树后也会大大提高停用词过滤的效率。如果使用 STL 容器存储大小为 m 的停用词，则判断一个词是否是停用词需要进行 m 次的字符串比较。这样对于一篇含有 n 个词的文档，其时间复杂度为 $O(m * n)$。如果采用双数组 Trie 树进行存储，由于是对整个词的完全匹配，且词语已经过切分，所以在进行模式匹配时不再需要回溯，即判断一个词是否是停用词，需要进行的字符串比较次数仅与词语本身

的长度相同。因此，对于一篇含有 n 个词，平均词长为 $a(2\sim4)$ 的文档，其时间复杂度为 $O(an)$。

7.3.4 关键词提取

除采用常用的 TF-IDF 特征外，将词的邻接类别(accessor variety，AV)值、词的空间局部性、词所在句子的位置作为非常重要的因素加入到词的权重中。

1. 词的 TF-IDF 特征

TF-IDF 的常见计算公式为

$$w(t,\vec{d})=\log(\mathrm{tf}(t,\vec{d})+1.0)\times\log(N/n_t+1.0)$$

其中，$w(t,\vec{d})$ 为特征词 t 在文档 $\vec{d}$ 中的权重；$\mathrm{tf}(t,\vec{d})$ 为特征词 t 在文档 $\vec{d}$ 中的出现频率；N 为文档总数；n_t 为文档集合中出现 t 的文档数量。

在自动文摘提取中，有时由于文档数量一般比较少，甚至在采用单文档文摘的方法中文档数为 1，因此 idf 区分度不大。Neto 等(2000)在此基础上提出用逆句子频率 ISF 来代替 IDF，进行句子级的词项权重度量，实质上仍属于 TF-IDF 的词项权重计算方法。由于句子非常短，包含的有效词数量很少，不能单独表达完整的意思，使用 ISF 的效果并不理想，因此仅取词频作为其 TF-IDF 特征。

2. 词的邻接类别值

在现实社会中，经常会有这样的人出入于不同的场合，与每个场合中的人打交道，有这样或那样的联系。在我们所观察的这个社交圈内，通常认为这样的人非常活跃，他作为这个圈子的代表性比其他人都重要。词的邻接类别值就是与社会网络这个原型中类比而应用于词的权重计算。Gauch 等(2000)曾使用词上下文向量表示该词的同义词。Feng 等(2001)使用邻接类别的概念来描述字符串的使用灵活性。把一篇文章比作社会网络中的社交圈，而构成文章基本单位的词就可以比作社交网络中不同的人。根据前面的叙述，一个词出现的语境越多，则说明这个词在整篇文章中越具有重要的意义。

这里用该词前后出现的 n 个词表示词出现的语境(词上下文)。字符串的左邻接类别是指在该串左边出现的字或词的种类数量。同样，字符串的右邻接类别是指在该串右边出现的字或词的种类数量。字符串的总邻接类别则定义为其左边和右边出现的字或词的种类数量。词的邻接类别越大，表明其使用越灵活，其在不同语境中使用的几率越大，因而其成为关键词的概率越大。例如，下面一段文字。

8 月 1 日，北京市召开迎**国庆交通**安全工作会。北京市交管局将对参与**国庆**活动的车辆进行严格检查，同时严厉打击闯红灯、酒后驾车等违法行为，确保**国庆**期间**交通**环境安全稳定。

北京市公安局公安交通管理局局长宋建国说，将对所有参加**国庆**活动单位的车辆，进行严格检查，确保安全性能良好。同时，与所有参加庆典的驾驶员进行面谈，强化安全意识，使其遵守**国庆交通**安全规定，杜绝**交通**违法和**交通**事故。

如果将上下文窗口的 n 设为 2，则字符串“国庆”的左邻接集合去掉单字词为{召开、参与、确保、参加、所有、使其、遵守}，右邻接集合为{交通、安全、活动、期间、单位}，因此字符串“国庆”在该语料上的左邻接类别为 7，右邻接类别为 5，总邻接类别为 12。类似的，交通的左邻接类别为 5，右邻接类别为 5，总邻接类别为 10。

从上面的例子可以看出，采用邻接类别对于发现词频比较高的关键词比较有效。

3. 词的位置局部性

词的频率、邻接类别值可以有效提取出词频比较高的关键词，但却很难有效提取出词频较低的关键词。张庆国等(2006)曾使用词语直径、词语分布偏差等特征提取关键词。词语直径是指词语在文本中首次出现的位置和末次出现的位置之间的距离。词语分布偏差考虑的是词语在文章中的统计分布。使用词在文档中出现位置的方差来表示词的位置局部性。假设候选词 T 在语料中出现 n 次，出现位置分别为 $P_1, P_2, \cdots, P_n$，则 T 的位置方差 $D(T)$ 为

$$D(T)=\frac{\sum_{i=1}^{n}(P_i-\overline{P})^2}{n-1}$$

其中，$\overline{P}$ 表示候选词 T 位置的均值。

词的位置方差表示词的分布稀疏程度。位置方差越小，则各个位置集中程度高，即该字符串的局部性强。因此，$D(T)$ 越小，局部性越好；反之，局部性越差。假设用 LE(locality estimation)表示字符串的局部性度量值，则局部性为

$$\mathrm{LE}(T)=1/D(T)$$

4. 词所在句子的位置

调查结果显示，段落的论题是段落首句的概率为 85 %，是段落末句的概率为 7 %。因此，采用分段函数量化句子位置对词权重的影响，即

$$\mathrm{SP}=\begin{cases}-\alpha(x-l/2), & x<l/2\\ \beta(x-l/2), & x>l/2\end{cases}$$

其中，x 表示句子在段落中的位置；l 表示段落中句子的数量；α 表示段落上半段中句子位置的平滑因子；β 表示下半段中句子位置的平滑因子。

5. 词的权重计算

词的最终权重根据上述的三个特征加权求和得到。各个特征的加权系数通过训练数据集人工训练得到。

7.3.5 句子切分

在词法分析阶段，文章中的标点符号被标注为/w 类别且做了切分，在扫描切分标注过的文本串时，遇到词性为/w 的片段，判断是否符合切句规则，符合则在此断句。如果一个标点符合如下的条件，则进行断句。

(1) 标点为“。”、“?”、“!”、“……”。

(2) 标点为“，”、“；”、“(”、“)”、“""”、“——”，并且句子开始到标点的距离大于句子长度阈值。

对于没有标点的段落结束位置，自动添加句号，并从该位置断句。另外，由于网络文档内容的复杂性，对于没有标点的长句，设置一个最大阈值，如果超过该阈值，则对该句子进行进一步处理。

(1) 如果前面的内容为重复的单字(切分后都是单个字)组成的串，则丢弃。

(2) 如果前面的内容全部为非重复的单字(切分为一个一个的字，不成词)串，则丢弃。

(3) 向前寻找用于句子结尾的语气助词，如“了”、“吗”、“等”等，如果有，则从该位置断句；如果没有，则丢弃该句子，从该位置继续扫描。

7.3.6 句子相似度计算

将句子视为向量，向量的每一维代表句子的一个片段，该片段的粒度可以是单个的字、词或短语、n 元组等，统称为词项，用 t 表示；向量的每个元素代表词项的权重，用 w 表示。这样句子 S 可以表示为向量$(t_1, w_1; t_2, w_2; \cdots; t_N, w_N)$，词项 t 称为特征，其权重 w 称为特征权重，简记为 $S(w_1, w_2, \cdots, w_N)$。句子表示为向量空间模型，需要确定特征及特征权重，我们采用词语作为特征，权重的影响因素有很多，选择 TF-ISF 作为其主要计算依据。最简单的方法是计算句子向量之间的内积，即

$$\mathrm{sim}(Q,D) = \sum_{t \in Q \cap D} w_{d,t} * w_{q,t}$$

文本处理中常用的相似性度量方式是向量的夹角余弦函数，即

$$\mathrm{sim}(Q,D) = \frac{\sum_{t \in Q \cap D} w_{d,t} * w_{q,t}}{\sqrt{\sum_{t \in Q} w_{q,t}^2 * \sum_{t \in D} w_{d,t}^2}}$$

候选文摘句与文摘句之间的句子冗余度采用余弦相似度表示。一个候选文摘句 s_i 与文摘句子集合 S 由两部分组成：一是候选文摘句与文摘句子集合中句子间的最大冗余度 $R_{\max}=\max_{s_j\in S}\mathrm{sim}(s_i,s_j)$；二是候选文摘句与文摘句子集合中所有句子的平均冗余度 $\bar{R}=\dfrac{\sum_{j=1}^{n}\mathrm{sim}(s_i,s_j)}{n}$，其中 n 为 S 中句子的总数。所以候选文摘句的权重重新计算为 $W'=\lambda W-(1-\lambda)(\alpha R_{\max}+(1-\alpha)\bar{R})$，其中 W 是仅根据句子中词的权重计算出的句子权重，在多次计算句子冗余度时不发生改变；λ 和 α 是调节系数，一般通过一定的训练集训练得到。

计算出候选文摘句权重，对所有的候选文摘句重新排序，找到权重最大的候选句，判断是否符合文摘长度要求，如符合，则将该句加入文摘句子集合；如不符，将该句标记为非候选句，取第二个候选文摘句进行判断。

7.4　面向主题的自动摘要

在信息检索领域，文档内容相关性作为文档检索中的主要度量方法，在过去的几十年时间里得到了深入研究。文档可以是信息内容任何一种层次上的粒度，如文档集、文档、段落、句子等。在自动文摘的任务中，这里的文档可以直接用句子来表示，于是相关度就变成句子与查询之间的相似度。

文档摘要除了要反映文档的主要内容，在信息检索领域还要与查询相关，这样才能提供用户真正关心的信息。用户提交的一个查询可以看做是一个主题，根据查询提取的摘要称为面向主题的摘要。面向主题的文档摘要系统架构如图 7.4 所示。

在对查询主题的描述中，停用词处理之后的词语可以分为辅助词和查询词两类。查询词是真正表达用户查询意图的词语。辅助词虽然能够独立地表达实际的意义，但是在主题中仅仅起到使语气连贯、礼貌客套等辅助作用，对实际要查询的内容并没有实际意义，去掉这些词，并不影响人们对查询真正表达语义的掌握。例如，查询语句“麻烦帮我查查《中国农民调查》这本书。”，真正的查询请求是“中国农民调查”，而“麻烦”、“帮”与“查”等词都是辅助词。

有关上述问题，可以通过定义查询句模加以解决，相关工作及其进展可参阅相关文献(高凯，2010)。

7.4.1　改进的最大边缘相关度方法

Carbonell 等(1998)提出用边缘概念进行文摘句的选取，并基于边缘度概念提出一种最大边缘相关度的文摘方法。其本质是在进行文摘句选取时，使候选文摘

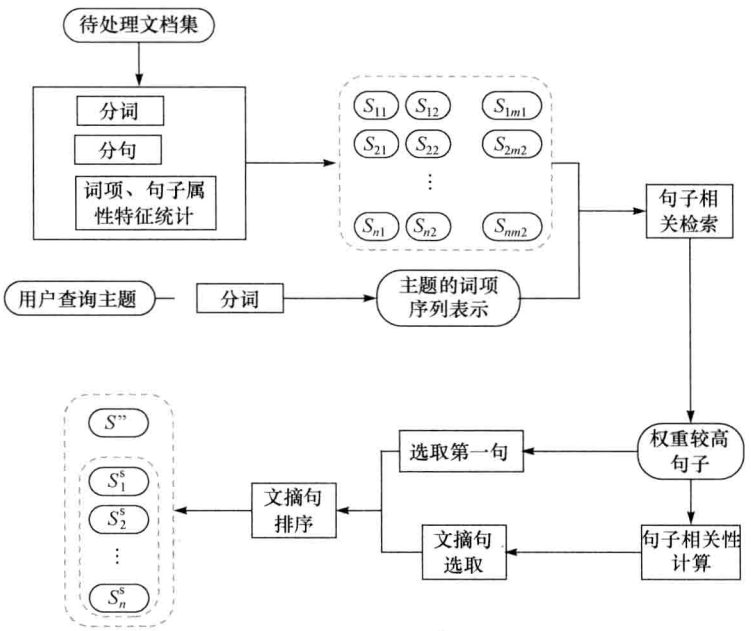


图 7.4　面向主题的文档摘要系统结构

句中要选入文摘句集合的句子既和主题的相关度较高，又使该句与已选文摘句集合之间的冗余度尽可能小，以此来保证句子与查询间的相关度，同时又减少文摘的冗余信息，增加内容的覆盖面，从而得到较高的文摘质量。计算方法为

$$\mathrm{MMR}=\mathrm{argmax}_{s_i\in(D-S)}\{\lambda \mathrm{sim}_1(s_i,q)-(1-\lambda)\max_{s_j\in S}\mathrm{sim}_2(s_i,s_j)\}$$

其中，q 表示查询主题；D 表示由句子集合组成的文档；S 表示已选文摘句集合，是 D 的子集；D-S 表示集合 D 与 S 的差；s_i 表示候选文摘句；sim_1 表示候选文摘句与查询的相关度；sim_2 表示候选文摘句与已选文摘句子集合之间的冗余度；参数 λ 用来调整这两个因素的权重。

最大边缘相关度方法能够较好地获取与查询相关的摘要，但是难以反映文档主要内容。因此，需要将句子本身的权重考虑进来，则公式变为

$$W_{q,s_i}=\alpha W_{s_i}+(1-\alpha)[\lambda \mathrm{sim}_1(s_i,q)-(1-\lambda)\max_{s_j\in S}\mathrm{sim}_2(s_i,s_j)]$$

s_i 与 s_j 之间的冗余度可以表示为最大冗余度与平均冗余度加权求和，这样更能体现出候选文摘句与文摘句子集合之间的冗余度。

7.4.2　面向主题的词特征统计

相对于基于关键词提取的摘要来说，面向主题的摘要是检索内核的组成部分，因此可以像检索一样调用每个词的倒排链表获取其统计信息。由于检索索引中存储的是所有文档的词信息，这对于提取某篇文档或者子文档集的词特征信息比较

复杂，需要查找的空间非常大。可以借鉴倒排索引来存储摘要的分词结果，以便进行词频、词的位置、词上下文信息以及句子位置的统计。基于倒排链表的词特征统计对于长度较小的单文档来说效率提升不大，而对于长文档或者多文档来说，效率的提升比较大。

7.4.3　领域主题词表

对于某些特定的应用领域，例如针对汽车行业的垂直搜索引擎中摘要的提取，如果加入行业知识，应能提高摘要的质量。我们采用加载用户主题词表的方式增加行业术语的权重。主题词表的格式为“词 词权重”，下面是关于司法的主题词表的一部分。

政法 0.8
法制 0.6
法律 0.65
法院 0.5

主题词表用于词权重计算的最后一步，根据词的统计特征计算出一个权重 W_t 后，将用户主题词加权到该词的权重中，即

$$W'_t=\lambda W_t+(1-\lambda)W_u$$

如果仅在分词之后引入主题词表，则会出现意外情况：可能存在被切散的词片段，那么即使这些词片段组成的词出现在主题词表中，也毫无意义。因此，在分词阶段，就加载主题词表到分词词典中，应该不会再出现这种情况。

这样也为分词引入了领域词典，虽然会改进摘要的性能，但是由于摘要与检索使用的是相同的分词词典，所以会造成检索的问题。例如，在邮政组织机构搜索中输入“中国邮政集团公司”的查询结果如图 7.5 所示。

机构名称	行政区划	机构地址	邮编
中国邮政集团储蓄银行千阳支行宝平路行	陕西省宝鸡市千阳县	宝平路10号	721100
中国邮政集团公司贵州分公司望谟县打易镇邮政支局	贵州省黔西南布依族苗族自治州望谟县	打易镇打易村0号	552301
中国邮政集团公司贵州分公司望谟县邮政局	贵州省黔西南布依族苗族自治州望谟县	天马大道0号	552300
中国邮政集团公司厦门邮政	福建省厦门市思明区	仙岳路458号	361012
中国邮政集团公司邮政储汇局	北京市西城区	金融大街甲3号	100033
中国邮政集团公司信息技术局	北京市西城区	金融大街甲3号	100033
中国邮政集团公司	北京市西城区	金融大街甲3号	100033
POWER BY AddreSmart	首页 尾页 第 1 页/共 1 页 共 条记录	页	跳转

图 7.5　组织机构搜索结果

如果分词词典中含有“集团公司”这个词，那么上面查询结果中后面 6 条在创建索引时都会出现在词“集团公司”的倒排链表中。这时如果输入“中国邮政集团”

进行查询，将只能查出一条结果，从主观感觉或自然语言的角度来说，这样是不正确的。因此，在分词中引入主题词表时，需要谨慎考虑应用场合。

7.4.4 句子间的包含关系

句子之间可能存在相似性，一个句子与另外几个句子是包含关系的情况也比较常见。例如下面的例子。

> “中国未来 20 年房价上涨的压力仍然是很大的。”姜伟新说。
>
> 中央政府抑制房价过快上涨的决心更大。
>
> 住建部部长姜伟新明确表示，中国未来 20 年房价上涨压力仍然很大，但中央政府抑制房价过快上涨的决心更大。

上例是通过句子切分得到的三个句子，其中第三个句子包含了第一、二个句子。我们将第一、二个句子称为第三句的子句，第三个句子称为长句。在子句的长度很小的情况下，仅采用余弦相似度计算得到的子句与长句之间的冗余度会比较小，无法区分它们之间的包含关系，因此需要另外的方法来表征这种关系。可以定义一种相对的词重叠度概念，将两个句子分别看成词集合 A 和 B，其中 A 在 B 之前，最简单的信息冗余度可以量化为 $|A \cap B|$，则

$$\text{Overlap}B_A = \frac{A \cap B}{B}$$

$\text{Overlap}B_A$ 表示句子 A 和 B 相对于 A 的重叠度。因此，句子 A 和 B 之间的包含度可以定义为 $\text{Contain}(A,B) = \max(\text{Overlap}B_A - \text{Overlap}A_B)$。将上例中三个句子分别看成词集合 A、B 和 C，计算第一、三句之间包含关系的过程如下。

(1) 计算出 A、C 之间相对于 A 的重叠度 $\text{Overlap}CA$。

(2) 计算 A、C 之间相对于 C 的重叠度 $\text{Overlap}AC$。

(3) 计算句子 A、C 之间的包含度 $\text{Contain}(A,C) = \max(\text{Overlap}CA, \text{Overlap}AC)$。

(4) 若 $\text{Contain}(A,C)$ 大于等于预定义的包含度阈值 $\text{Contain}(\text{Thresh})$，则 C 包含 A（$\text{Overlap}CA$ 大于 $\text{Overlap}AC$）或 A 包含 C（$\text{Overlap}AC$ 大于 $\text{Overlap}CA$）；否则，A 与 C 之间没有包含关系。

在课题组于 2009 年 12 月 4 日采集的 245 890 篇新闻语料中，共抽取出19 048 对具有包含关系的句子。从中随机选取 100 对句子进行人工评测，其中有 28 对没具有包含关系，在这个小测试集中得出的准确率为 72%。下面是其中的几个例句。

> 广州市副市长、亚组委常务副秘书长许瑞生表示，还没开始拍卖，不能说就是给了万科。

没开始拍卖，怎么说是给了万科？

程卫东说，范龙佩的当选并不是个人能力有多高，而是其背后的大国和小国之间的利益之争的结果。

范龙佩当选是各方平衡的结果。

事实上，马琳主动提出来分割给女方的，比按照法律规定判下来的还要多一些，但女方一直不满意。

男方：主动给女方的已比规定的多。

据吴江路东段管委会介绍，本月底吴江路东段将拆迁完毕。

记者从吴江路东段管委会获悉，本月底，吴江路东段将拆迁完毕。

在前面所述的包含度计算过程中，句子中每个词都赋予相同的权重 1，这时如果子句比长句增加或减少几个一般意义的词，并且这些词不是停用词，就会产生比较大的噪声，难以计算出它们之间的包含关系。因此，如果在计算重叠度时为各个词赋予不同的权重，就可以有效去除增加的噪声。带权重的词重叠度定义为

$$\mathrm{Overlap}B_A = \frac{\sum_{t \in A \cap B} w_t}{\sum_{b \in B} w_b}$$

其中，w_t 表示在句子 A、B 中同时出现的词 t 的权重；w_b 表示出现在句子 B 中的词 b 的权重。

7.5　实验与分析

测试语料为 2009 年 12 月 4 日从互联网中采集到的新闻文章正文，正文总大小为 466 487 151 字节，文章数目为 246 177 篇。

7.5.1　稳定性测试

对于每一篇文章的标题和正文，调用基于关键词提取的自动摘要组件提取摘要，共提取出摘要 245 890 篇，丢失 287 篇。这 287 篇摘要丢失原因包括源文档内容仅包含无意义字符、源文档乱码、源文档为纯英文。

7.5.2　时间性能

对于测试语料，选择其中一部分（208 篇）文档，统计获取摘要需要的时间，结

果如图 7.6 所示。

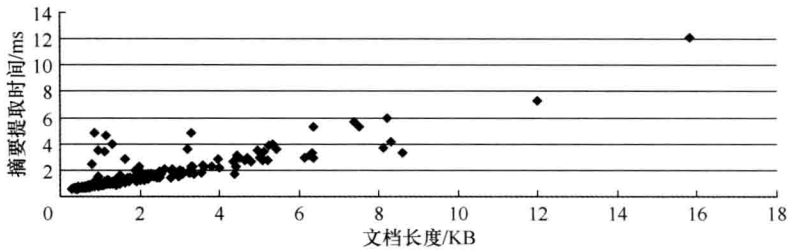


图 7.6 文档长度-摘要提取时间散点图

图 7.6 中部分点的数据如表 7.2 所示。

表 7.2 文档长度-摘要提取时间数据

文档长度/KB	0.379	0.392	0.401	2.166	2.214	2.235	3.804	4.003	11.996	15.834
时间/ms	0.621	0.55	0.575	1.436	1.717	1.591	2.269	2.203	7.277	12.069

根据统计结果，绝大多数文档长度都小于 6KB，摘要的提取时间一般控制在 4ms 以内，摘要的提取时间随着文档长度的增加而增长。

1. 标准垂直搜索系统检索性能测试

针对国标文件的标准垂直搜索的测试主要分为准确率召回率性能、响应时间测试。

由于标准号检索、标准详细信息检索的规则十分复杂，而且标准号的命名规则时有不规范的情况，因此有必要对垂直搜索的准确率、召回率进行测试。准确率、召回率的测试过程为选择测试用例；将测试用例分别提交给数据库、搜索引擎进行查询；以数据库结果为标准，统计结果的准确率、召回率。

测试用例要覆盖到不同类型的标准号，以及包含不同格式引用标准的情况，由用户提供。因此，测试用例比较庞杂，在此不再展开叙述。准确率、召回率测试经过多次改进后通过。部分标准号用例如表 7.3 所示。

表 7.3 标准号测试用例

GB 10069.3-2008	14 条引用标准
GB 10068-2008	引用标准 gb997 模糊查询结果有 708 条
ANSI/ANS-3.4-1996	标准表里不存在该标准号，类似的是“ANSI/ANS 3.4-1996”
MIL-T-10010H-1989	该标准号不存在，类似的也没有

2. 响应时间测试

测试数据库中共有 360 934 条记录，每条记录共 8 个字段。平均每条记录总大小为 2K 左右。在 3M 小区宽带的空闲网络环境下，建索引平均耗时 347.2s。

在 CPU 为 2.6GHz、内存 2G 的 PC 机上开启检索服务，通过客户端提交查询，其搜索的响应时间如表 7.4 所示。

表 7.4　无摘要搜索响应时间

查询词	响应时间/s
GB 2312	0.168
GB 1	0.281
GB 12	0.24
2312	0.078
10068	0.094
1	0.328
石油	0.015
信息交换	0.016
维吾尔 哈萨克	0.009
信息技术	0.016
导轨	0.018

通过以上测试，可以看出该垂直搜索对中文的检索时间最快，一般在十几毫秒左右；对数字的检索由于精确匹配的处理，响应时间较慢；对既有数字又有字母的标准号检索响应时间最慢，这是由于含有字母的标准号检索时需要将序号切分出来进行精确匹配导致的。

将面向主题的摘要加入垂直搜索系统后，每次返回 20 条结果，即最多计算 20 篇文档的摘要。响应时间如表 7.5 所示。

表 7.5　包含摘要的搜索响应时间

查询词	响应时间/s
GB 2312	0.187
GB 1	0.235
GB 12	0.26

续表

查询词	响应时间/s
2312	0.078
10068	0.103
1	0.305
石油	0.098
信息交换	0.106
维吾尔	0.089
信息技术	0.134
导轨	0.083

可以看出，由于加了摘要提取，对返回结果较多的内容检索响应时间影响较大，而对于返回结果很少的标准号检索则影响不大，响应时间甚至因为机器负载的略微不同而降低。

7.5.3 文摘质量

文摘质量的评测是自动文摘研究的难点，目前仍面临许多挑战。文摘系统通常按照不同的压缩率生成相应文摘，在不同压缩率下，文摘包含的原文档中的信息量是不同的。如何在评价中反映这一变化，增加了自动评价的难度和复杂性。文摘通常是面向需求的，应该根据用户或应用的需求包含特定的信息。如何在进行评价时也对用户或应用的需求加以考虑，这也使自动评价变得更加复杂。

人工评测需要投入大量的人力、物力，因此我们仅选取了少量 14 篇文档来人工考察文摘的质量，结果表明系统能够有效提取出文摘，根据开放测试版本中用户的反馈来改进文摘的质量。

7.6 自动摘要应用场景分析及大数据搜索与挖掘软件应用示例

LJParser 大数据搜索与挖掘开发平台中提供摘要及关键词提取功能，其中自动摘要能够对单篇或多篇文章，自动提炼出内容的精华，方便用户快速浏览文本内容。关键词提取可以对单篇文章或文章集合，提取出若干个代表文章中心思想的词汇或短语，可用于精化阅读、语义查询和快速匹配等。本例选取两个语料库，一个是关于经济的短篇文档集，一个是中国四大名著中的《西

游记》长文本。对两个语料库的自动文本摘要和关键词提取如图 7.7 和图 7.8所示。

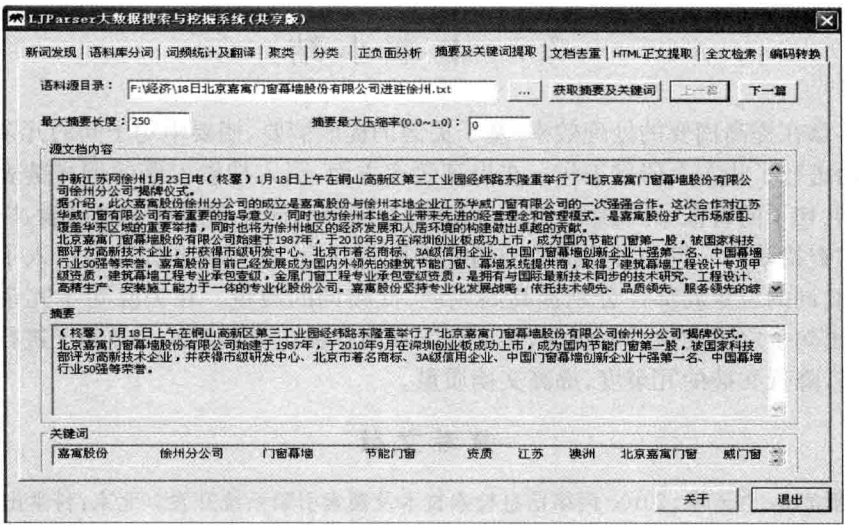


图 7.7　对短文本的自动摘要和关键词提取

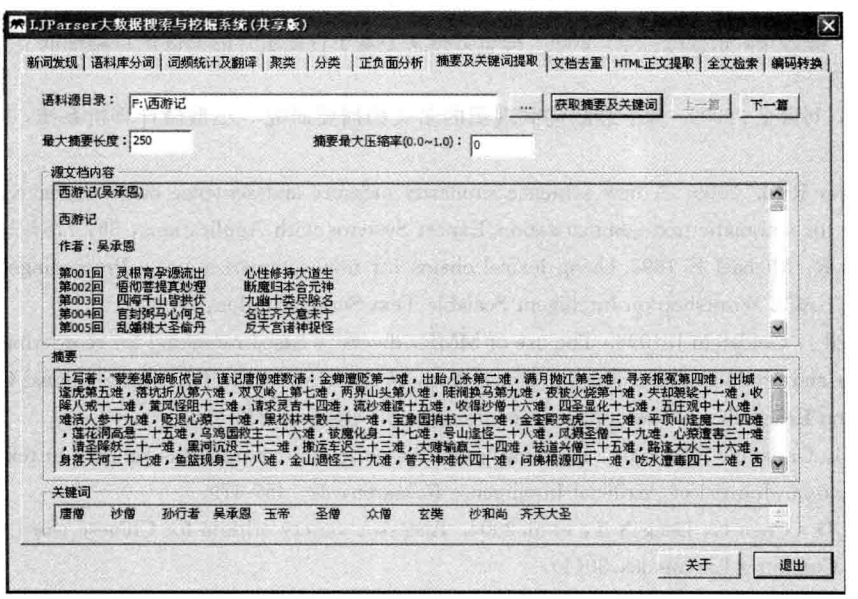


图 7.8　对长文本的自动摘要和关键词提取

可以看出，LJParser 的自动摘要和关键词提取功能对短文本和长文本的效果都是非常好的。

7.7 本章小结

本章在提高摘要的处理效率、基于关键词提取摘要、摘要中句子间的冗余去除等方面进行了相关个研发工作。在提高效率方面，引入搜索引擎的倒排链表结构统计词、句子的特征，并使用双数组 Trie 树存储分词词典和用户主题词表，以期提高词的查找效率。将关键词提取与摘要提取相结合，在关键词提取中引入了词邻接类别、词的位置局部性分别提高高频词、低频词的质量。在去除句子冗余度方面，提出句子之间的包含度概念，能够对文章中存在包含关系的一类句子进行有效的排重，降低文摘的冗余度，提高文摘质量。

参考文献

高凯，郭立炜，许云峰．2010. 网络信息检索技术及搜索引擎系统开发．北京：科学出版社，1-20.

高凯．2010. 基于句模的归约算法在自然语言检索中的应用．情报学报，29(1)：22-31.

刘挺，王开铸．1999. 自动文摘的四种主要方法．情报学报，18(1)：10-19.

张庆国，薛德军，张振海，等．2006. 海量数据集上基于特征组合的关键词自动抽取．情报学报，25(5).

周进华，刘贵全．2009. 基于衰减词共现图的多文档摘要研究．小型微计算机系统，30(1)：173-177.

Aliguliyev R M. 2009. A new sentence similarity measure and sentence based extractive technique for automatic text summarization. Expert Systems with Applications, 36: 7764-7772.

Barzilay R, Michael E. 1997. Using lexical chains for text summarization// Proceedings of the ACL/EACL Workshop on Intelligent Scalable Text Summarization: 86-90.

Carbonell J, Goldstein J. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries// Proceedings of the 21st ACM-SIGIR International Conference on Research and Development in Information Retrieval: 335-336.

Erkan G, Dragomir R R. 2004. Lexrank: graph-based lexical centrality as salience in text summarization. Journal of Artificial Intelligence Research, 22: 457-479.

Feng H D , Chen K, Deng X T, et al. 2004. Accessor variety criteria for Chinese word extraction. Computer Linguistics,30(1).

Gauch S , Wang J. 2000. Corpus analysis for TREC-5 query expansion// Proceedings of the Fifth Text Retrieval Conference(TREC-5): 537-546.

Jones K S. 1995. Introduction: automatic summarizing. Information Processing & Management, 31(5):625-630.

Luhn H. 1958. The automatic creation of literature abstracts. IBM Journal of Research and Development, 2(2):159-165.

Neto J L , Santos A. 2000. Generating text summaries through the relative importance of topics // Proceedings of IBERAMIA-SBIA: 300-309.

Radev D R, Jing H Y, Magorzata S, et al. 2004. Centroid-based summarization of multiple documents. Information Processing & Management, 40(6): 919-938.

Sparck K. 2007. Automatic summarizing: the state of the art. Information Processing & Management, 43(6):1449-1481.

Zhang J, Cheng X Q, Wu G W, et al. 2008. AdaSum: an adaptive model for summarization // Proceedings the ACM 17th Conference on Information and Knowledge Management.

第 8 章　JZSearch 大数据精准搜索引擎

8.1　概　　述

JZSearch 大数据精准搜索引擎是针对大数据垂直搜索需求的全文检索引擎，先后研制 3 年多，已经实际应用到中国邮政、中国标准搜索、微博搜索、新疆维吾尔舆情搜索等多个系统。整个系统内容基于 C++开发，支持文本、数字、日期、字符串等各种数据类型，多字段的高效搜索，支持 AND/OR/NOT 以及 NEAR 邻近等查询语法。同时，支持维语、藏语、蒙语、阿拉伯语、韩语等多种少数民族语言及方言的检索，可以无缝地与现有文本处理系统与数据库系统融合。其主要特性如下。

(1) 可以按照任意指定字段进行排序，支持指定字段的搜索，也可以搜索多个字段，完成复杂表达式的综合搜索。

(2) 支持精确匹配以及模糊匹配，默认为精确匹配。

(3) 内嵌正负面情感等极性分析，也可支持类别搜索。

(4) 支持语义联想搜索，如搜索“马铃薯”可以同时返回“土豆”的内容，搜索“北京市”可以返回“北京”或者“首都”的内容；用户可以根据业务需要进行语义联想词表定制。

(5) 支持增量索引，系统可以在搜索服务不中断的前提下，继续索引新的数据。

(6) 具有自动备份与恢复机制，在建立索引和自动优化之前，系统会将已有的索引文件自动备份；在当前索引文件被破坏无法搜索时，系统将自动恢复上次正常的备份文件。

(7) 具有自动缓存机制，系统自动保存最近常用的搜索条件与结果，再次搜索时将直接推送搜索结果内容，可以将搜索响应速度提升 30%以上；缓存会随着新的索引数据自动更新，不存在缓存延迟问题。

(8) 提供自动优化机制，在系统索引碎片较多时，系统会自动优化归并。

(9) 支持多线程搜索服务，兼容当前所有厂商的数据库系统，如 SQL Server、Oracle、MySQL、DB2 等；支持 Windows、Linux、FreeBSD 等操作系统；支持C/C++、C#、Java 二次开发。

8.2　JZSearch 大数据搜索引擎系统架构

JZSearch 大数据精准搜索引擎的系统架构如图 8.1 所示。

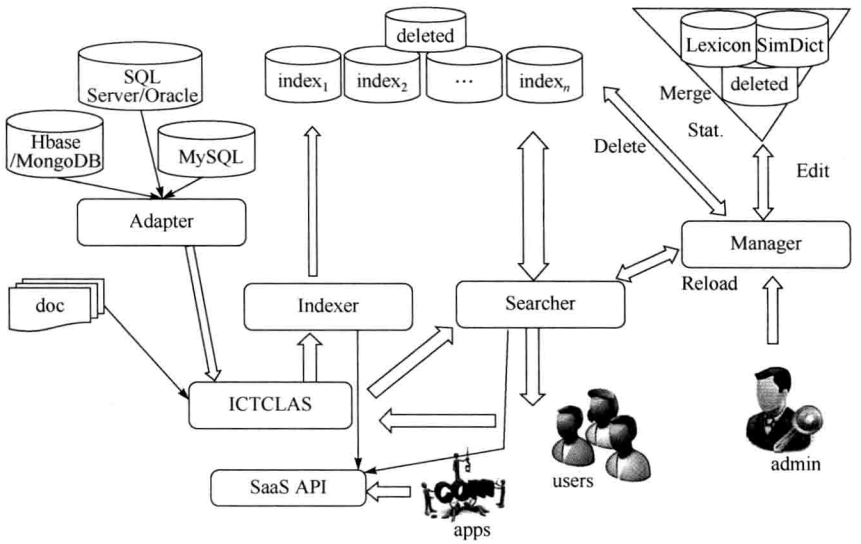


图 8.1　JZSearch 大数据搜索引擎架构

整个系统包括核心组件有索引器 Indexer、搜索器 Searcher、内部配套的词法分析工具 ICTCLAS、搜索管理器 Manager；外围组件有数据库适配器 Adapter、供开发人员调用的 SaaS API。各部分主要功能如下。

(1) Inderxer(索引器)。

对各类结构化与非结构化数据进行高效索引，索引文件以索引桶的形式存储。

(2) Searcher(搜索器)。

对用户查询进行解析，并搜索所有的索引桶，按照排序算法要求输出各类形式的查询结果并返回给用户。

(3) ICTCLAS(词法分析工具)。

对用户查询与文档进行词法分析，提供分词结果为后续的计算提供词法粒度，可以解决汉语与英语的编码转换与词法处理。少数民族语言搜索的词法分析采用 n-gram 进行索引。

(4) Manager(搜索管理器)。

为管理员提供精准搜索引擎的后台知识与索引文件，提供管理工具，搜索管理器采用命令行、可视化界面及 API 的形式，实现同义与相关词扩展、索引自动归并、部分索引内容的删除、索引项自动统计等功能。

(5) Adapter(数据库适配器)。

自动对已有的数据库进行扫描遍历，实现对数据库的无缝支持，从而为数据库管理系统提供全文搜索功能，目前支持的数据库系统包括 Oracle、SQL Server、

MySQL 以及新型的 HBase、MongoDB 等数据库，从而为各类业务系统提供结构化与非结构化数据的联合搜索服务。

(6) SaaS API(供开发人员调用的精准搜索引擎服务 API)。

JZSearch 的索引与搜索系统可以自动搭建，针对已有的业务数据提供无值守的搜索服务。开发人员只要熟悉搜索语法，采用 SaaS API 即可使用搜索服务，嵌入自己的应用程序，实现大数据的精准搜索。

JZSearch 大数据精准搜索引擎提供的是在线增量索引功能，在整个搜索服务不间断的情况下，实现增量索引，索引完成后自动加载增量数据；如果索引桶碎片过多，则自动对索引桶进行归并操作。为实现各种不同状态的自动切换，JZSearch 采用文件 JZSearch. status 对状态进行管理。具体的状态转换如图 8. 2 所示。

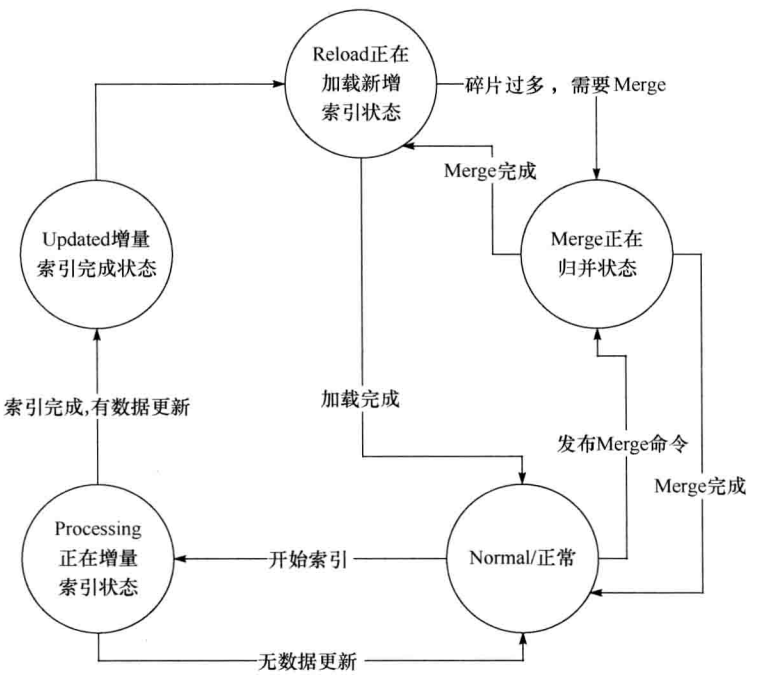


图 8. 2 JZSearch 大数据搜索引擎的状态转换图

8. 3 JZSearch 索引关键技术

8. 3. 1 索引字段类型

JZSearch 兼容当前基本的数据类型，如表 8. 1 所示。

表 8.1　JZSearch 索引字段类型

类型名	类型值	说明
FIELD_TYPE_TEXT	1	文本类型
FIELD_TYPE_INT	2	整型(32 位整型数据)
FIELD_TYPE_LONG	3	长整型(64 位整型数据)
FIELD_TYPE_DATETIME	4	日期时间型(目前等同于 FIELD_TYPE_INT)
FIELD_TYPE_FLOAT	5	浮点型
FIELD_TYPE_BIGTEXT	6	存储在大文件中

8.3.2　索引词项的设计

JZSearch 采用固定的静态索引词项，而不是动态变化的词典。在中英文搜索引擎的词典方面，它包括了 8 万中文词语以及 7 万左右的常用英文词语；静态词典采用高效的查询算法；采用双数组 Trie 树词典管理与检索方法，可以实现每秒 100 万次的查询速度，算法时间复杂度仅与被查询词条的长度相关，而与词典规模无关，内存占用一般仅仅在 2MB 以内。动态词典的查询效率不到静态词典的 1/10，而词典的查询在一般的信息处理中将占据 90%以上的执行频率；同时过大的搜索词项将导致索引的急剧膨胀。静态词典解决了未收录词的处理，具体包括下面三种情况。

1）英文的特殊形式处理

处理英文不规则动词的不同时态、名词的单复数等情况，JZSearch 并不采用常见的词干还原的方法，这种方法过于简单，信息丢失严重。我们收集了不同形态的变换对应关系，直接输出对应的词原型，绝大部分原型都在索引词典中收录。

2）未收录中文词的处理

ICTCLAS 分词系统会产生系统未收录的人名、地名、新词等，采用索引词表对未登录词进行二次最大匹配，确保每个词都在索引词范围内，如“张华平”将按照“张”、“华”、“平”三个词分别索引。

3）未收录英文数字的处理

采用交叉三元组(每个三元组都已经收录到索引词典中)的形式分别索引，例如需要索引未登录的“ABCDE”，则依次索引项“ABC”、“BCD”、“CDE”；可以确保搜索 ABCDE 的任何两个以上字符的子串都可以查询到。单个或者两个字符的索引都是直接搜索，三元组可以确保搜索的效率。

8.3.3　索引压缩技术

倒排索引文件研究的一个关键目标是通过优化的算法来减少 I/O 带宽和存

储开销。索引文件的大小决定其所占用的存储开销，大索引文件需要更大的 I/O 带宽来读取，因此其大小也直接影响了处理时间。

尽管文本压缩已被广泛研究(Bell et al.，1990；Grossman，1995)，但是在压缩倒排索引文件方面的研究工作相对来讲还是比较少的。在 Moffat 和 Zobel 的研究中，他们使用了一种相对比较容易解压缩的索引生成方式。压缩索引后的大小不到原始文档集的 10%，而且这里面还包含了停用词。

尽管有些研究表明，假设内存代价相对较低，我们可以不必关注于索引压缩，但仍然需要做些工作(Witten et al.，1999)。《国王詹姆斯圣经》(大约 5MB)，包含了 9020 个词项，而传统的 TREC 文档集(稍多于 2GB)包含 538 244 个词项。当遇到新的领域时，新词的数量总会有一些增长，但是认为词表数量大约稳定在一百万或两百万个还是合理的。对于每一个索引词，索引词的平均长度为 6，文档频率计数器需要 4 个字节，倒排表入口的指针需要 4 个字节，所以总共需要 14 个字节。我们保守估计有两百万个索引词，未经压缩的索引很可能在 32MB 以内。甚至，我们可以上升一个数量级，存储索引所需的内存空间保守地讲也会在 1GB 以下。

假如索引相对较小，而且可以很容易加载到内存中，就不用过多地讨论索引压缩的技术细节了。我们注意到，词干还原降低了压缩的需求，相对直接的方法就是使用哈夫曼编码(Witten et al.，1999)。同样，使用短语可以提高召回率和准确率。按照短语存储索引也可以很好地压缩索引。这还取决于如何识别和限制短语，大多数系统往往会去除出现频率较低的短语。

在完成数值存储时，JZSearch 采用字节对齐(byte aligned，BA)的索引压缩算法(Grossman，1995)。通过使用字节边界来实现字节对齐压缩，运行时会付出些许代价，但可以获得一定的压缩比。字节对齐算法易于实现，而且压缩比比较高(使用停用词后，压缩文件大小占未压缩索引文件大小的比例大约是 15%)。变长编码压缩的具体介绍可以参见文献。尽管变长压缩可以获得更好的压缩比，但需要预先遍历索引内容，字节对齐算法与索引项无关，可以快速高效实现，在实际索引中更受推崇。索引压缩尽管需要一定的解压缩时间，但是将大大降低 I/O 时间。综合来看，索引压缩提高了索引和搜索的效率。

JZSearch 采用行程编码(ran-length enwding，RLE)方法用于存储索引的文档编号 DocID 与位置信息 Offset，这两部分的信息都是自动增量的。以文档 DocID 为例，对于任意一个文档 DocID，仅仅需要计算当前 DocID 以及待处理 DocID 之间的偏移量。例如，倒排表第 i 篇文档的编号为 DocID_i，倒排的下一个文档编号为 DocID_{i+1}，实际存储的时候 $i+1$ 个文档存储的值为 $\mathrm{DocID}_{i+1}-\mathrm{DocID}_i$，因此可以采用较小的值来存储。对于没有其他文档 ID 存在的情况下，就存储文档 ID 的压缩形式。采用这种技术，相对较小的数值可以保证在整个数据中占较大比重。这种方案可以有效减小 ID 的范围，允许它们以更精小的形式存储。

在数据压缩方面对于一个给定的输入值，保留最左面的两个比特，用来存储整个存储值占用的字节数。两个比特位的表现形式有四种可能的组合，因此对于所有的文档 ID，可以使用一个两比特位的指示器。整型数据的存储长度可以为 6 比特、14 比特、22 比特或者 30 比特。最佳的情况下，所有的值都小于 64，通过上述方法，每个单独的数据记录大小使用一个字节就可以存储。在没有压缩的情况下，所有的文档 ID 都需要使用 4 字节存储。

对于待压缩的每一个值，需要计算存储值所需的最小字节数。表 8.2 给出了可以存储的值范围。

表 8.2　整型数据的存储范围及其字节数

长度	所需字节数
$0 \leqslant x < 64$	1
$64 \leqslant x < 16\ 384$	2
$16\ 384 \leqslant x < 4\ 194\ 304$	3
$4\ 194\ 304 \leqslant x < 1\ 073\ 741\ 824$	4

给定任意一个索引词 t_1，考虑其索引的条目，假定 t_1 在文档 1、文档 3、文档 7、文档 70 和文档 250 中出现。字节对齐压缩使用最高的两个比特位来表明存储该值需要的字节数。对于前四个值，只需要一个字节就可以表示；对于最后一个值，需要两个字节来表示。需要注意的是，我们只需要计算倒排表中每一项的差，最后的差值为 250－70＝180。我们需要计算技术研究所有的最终数值。这些数值和压缩后的相应比特字符串如表 8.3 所示。

表 8.3　字节对齐方式压缩

值	压缩的比特串
1	00 000001
2	00 000010
4	00 000100
63	00 111111
180	01 000000 10110100

在没有压缩之前，倒排表中的每条记录需要 4 字节，5 条记录总共需要 20 个字节。这些数值和它们对应压缩后的比特字符串如表 8.4 所示。

表 8.4 对比的基准线:压缩前

值	未压缩的比特串
1	00000000 00000000 00000000 00000001
3	00000000 00000000 00000000 00000011
7	00000000 00000000 00000000 00000111
70	00000000 00000000 00000000 01000110
250	00000000 00000000 00000000 11111010

在这个例子中,未压缩的数据需要 160 比特,使用字节对齐压缩后仅需要 48 比特,压缩比为 48/160=30%。

8.3.4 内存交换

缓存技术是提高系统性能和可扩展性的一种重要手段,在计算机各个领域都有广泛应用。该技术最先应用于 CPU 设计中,由于内存的访问速度较 CPU 寄存器要慢几个数量级,然而 CPU 的存储容量有限,且比内存昂贵,所以将内存中的一部分内容缓存到 CPU 中,从而缓和 CPU 与内存之间的性能差距,提高系统性能。同样,将慢速的磁盘中的内容缓存到内存中,可以减少对磁盘的 I/O 操作,提高存取效率。常用的缓存替换算法有先进先出(FIFO)算法、最近最少访问策略(LRU)算法。

尽管目前服务器中配置的内存已经相当大,而且可以通过增加内存来满足 JZSearch 的索引需求的空间。但是这对一个应用系统来说,无疑是令人头痛的,而且检索系统的应用环境并没有那么简单。例如,与另外占用内存很大的系统同时运行在一台机器上,或者运行在一台配置比较低的 PC 机上。这些都要求 JZSearch 在内存的使用上有很强的伸缩性,所以索引缓存必不可少。

8.3.5 增量索引

在实际应用中,源文档集并不是一成不变的。例如,在针对国标文件的数字图书馆系统中,新增加标准、标准修订单、标准作废、标准实施等,每时每刻都可能有数据的增加、删除和数据内容的修改。对于这样的文档集,一般称之为动态文档集或动态文本。对于动态文本,如果采用静态的倒排索引,查询结果往往不能反映出最新的标准信息,新发布的标准无法出现在查询结果中,而已作废的标准状态没有改变。因此,倒排索引需要随着源数据的变化而更新。

向倒排索引中添加新的文档,需要抽取出文档中的词,再从词典中搜索出这些词的句柄或者索引顺序,然后根据索引文件找到该词的倒排链表,并在倒排链尾部添加该词的索引。然而,一篇文档往往包含几百到几千个词,每增加一篇文档就需

要对当前索引进行一次更新。为保证倒排索引的检索效率，每个词的倒排链在磁盘上是连续存放的。每更新一次倒排链，需要重新分配磁盘空间，然后将旧的倒排链数据拷贝到新的空间，并在尾部添加新的倒排链数据。这样会导致过多的磁盘读写以及磁盘拷贝操作，因此使用这种直接的方法往倒排索引中添加新索引数据是非常低效的。

Cutting 和 Pedersen 的研究表明，在倒排索引更新的过程中，将倒排索引缓存在内存中可以大幅度提高效率。每篇文档的添加首先在内存中进行，只有当内存缓冲区不够时，才更新磁盘上的索引数据，这样可以大大减少磁盘读写操作。这种方法需要定期地（一般是当内存缓冲区耗尽时）将内存中的倒排索引和磁盘上的倒排索引合并。

重新扫描文档集建立新的索引以替换旧的索引，是更新索引最简单直接的方法，这种方法称为索引重建（re-build）。这种方法的实现最简单，适合数据更新周期较长、对检索实时性要求不高的场景。但是对于持续不断有新文档增加（如通用搜索引擎中采集器不断采集到的新网页）的场景，该方法不但会导致频繁的磁盘读写，而且会有频繁的索引切换，这对检索器来说就有点捉襟见肘了。

第二种更新方法是单独更新每条倒排链，称为原地索引更新方法（in-place）。由于每个词的倒排链大小差异非常大，预先为每条倒排链分配多余的空间不但算法非常复杂，而且仍可能导致频繁的磁盘拷贝操作。

因此，我们采用第三种方法，这种方法是将内存索引和磁盘索引归并排序，形成新的磁盘索引，称为立即合并策略或再合并策略（re-merge）。这种方法不仅提高了新索引的构建效率，而且大大减少了磁盘读写、磁盘拷贝操作。这个方法分为两步，第一步是在内存中缓存索引，需要确定缓存的大小以及缓存写入磁盘的时机；第二步是索引归并，需要根据应用场景选择合适的归并策略。

8.3.6　数据库检索

针对数据库中的数据，索引器需要从数据库中读出数据，然后转换成适用于索引器的数据格式，最后对转换的数据建立索引。根据检索结果的呈现需求，将返回的字段内容存储到索引目录下。

ADO（activeX data objects）是基于组件的数据库编程接口，是一个和编程语言无关的 COM 组件系统。通过 ADO 访问数据库的方法如下。

（1）创建一个到数据库的 ADO 连接。

（2）打开数据库连接。

（3）创建 ADO 记录集 Recordset。

（4）从记录集读取需要的数据。

（5）关闭记录集。

(6) 关闭连接。

ADO Connection 对象用于创建一个到达某个数据源的开放连接。通过该连接,可以对一个数据库进行访问和操作。如果需要多次访问某个数据库,应使用 Connection 对象来建立一个连接,也可以由一个 Command 或 Recordset 对象传递一个连接字符串来创建某个连接。其主要的方法如下。

```
HRESULT CreateInstance();
//创建一个到数据库的 ADO 连接。
HRESULT Open ( _bstr_t ConnectionString, _bstr_t UserID, _bstr_t Password, long Options );
//打开数据库连接。第一个参数为数据库连接字符串。
```

ADO Recordset 对象用于容纳一个来自数据库表的记录集。一个 Recordset 对象由记录和列(字段)组成。在 ADO 中,此对象是最重要且最常用于对数据库的数据进行操作的对象。其主要的属性和方法如下。

```
_variant_t GetCollect ( const _variant_t & Index );
//从当前记录中返回一个字段,字段名作为参数传入,返回结果存放到
//_variant_t 变量中。
HRESULT CreateInstance(_uuidof(Recordset);
//创建一个结果集实例。
HRESULT Open ( const _variant_t & Source, const _variant_t & ActiveConnection, enum CursorTypeEnum CursorType, enum LockTypeEnum LockType, long Options );
//执行一条 SQL 语句,并将结果保存到创建的实例中。
adoEOF;
//表示结果集记录指针是否移到最后。
HRESULT MoveNext();
//记录指针移到下一个记录。
HRESULT Close();
//关闭通过 open 打开的结果集实例。
```

通过以上两个 ADO 对象,就可以从数据库中读取出所需要的记录及其内容。读取出来的记录内容需要转换成字符串格式,才能调用检索内核建立索引。这其中需要注意的是 BLOB 类型的转换,由于使用 SafeArrayAccessData 转换得到的字符串是从堆中申请的,所以在使用完该字符串后需要调用 SafeArrayUnaccessData 函数释放内存,否则将会造成大量的内存泄露。

在建立数据库连接时另一个关键的问题是数据的查询方式,即 SQL 语句的书写方式。由于我们使用的 ADO 访问数据库的方式不是一次性从数据库中读取出

所有记录(如果一次性读取,在数据库中数据量非常大的情况下,会导致内存耗尽),而 ADO 默认的方式是每次读取一定数量的结果放到缓存中,当需要的记录不在缓存中时,需要从数据库中读取。由于 ADO 的缓存机制不适合上文所述的应用场景,所以在查询时采用合适的分页机制是十分必要的。

8.4　JZSearch 搜索技术

8.4.1　JZSearch 排序算法

一般全文搜索引擎都是按照相关度排序的,排序算法有向量空间的向量相似度、OKAPI 算法、语言模型算法等,但是这远远达不到大数据实际的搜索需求。例如,我们需要从 Web 大数据中搜索“斯诺登”事件的最新动态,或者搜索附近最便宜的酸菜鱼,微博搜索中更需要综合考虑相关度与时效性。这些在目前的搜索引擎中都是做不到的。

JZSearch 提供多样化的搜索排序算法,也可以由用户自己定义排序的字段或者排序的顺序。排序命令的格式为[SORT] sort_type,用于设定当前搜索结果的排序方式,如表 8.5 所示。

表 8.5　JZSearch 排序类型

类型名	类型值	说明
docid	1	按照 docid 顺序排序,即先建索引的文档优先
relevance	2	按照相关度排序
reverse_docid	3	按照 docid 逆序排序,即后建索引的文档优先
<field_name> descend/ascend	4	按照指定的数值型字段进行排序

(1) [SORT]可以出现在搜索语句的起始、中间以及末尾。如果不指定的话,系统会按照系统配置的排序方式进行排序;不做任何指定的前提下,系统按照docid排序。

(2) sort_type 不区分大小写,只需要输入前五个字符即可。

(3) <field_name> descend/ascend field_name 必须是数值型,并设置为信息过滤。

(4) 在相关度排序方面,JZSearch 采取运算效率高的向量空间夹角余弦计算公式。

(5) 示例如下。

[FIELD] price [RANG] 1.0 9.0 [FIELD] name [AND] 牛奶 儿童 [SORT]

docid

价格在 1.0～9.0，且商品名包含“儿童”、“牛奶”，结果按照 docid 排序

[FIELD] price [RANG] 1.0 9.0 [FIELD] name [AND] 牛奶 儿童 [SORT] relev

价格在 1.0～9.0，且商品名包含“儿童”、“牛奶”，结果按照相关度排序

[FIELD] price [RANG] 1.0 9.0 [FIELD] name [AND] 牛奶 儿童 [SORT] rever

价格在 1.0～9.0，且商品名包含“儿童”、“牛奶”，结果按照 docid 逆序排序

8.4.2 JZSearch 结果格式

用户提交搜索后，搜索结果 XML 文件格式如下。

```
〈? xml version="1.0" encoding="gb2312" standalone="yes" ?〉
〈LJSearch-Result〉
  〈Result-Number〉
    〈Total-Number〉1〈/Total-Number〉//符合条件的搜索结果总数
    〈Return-Number〉1〈/Return-Number〉//返回符合条件的搜索结果总数
    〈Start-Position〉0〈/Start-Position〉//结果集合中的起始编号
  〈/Result-Number〉
  〈Result〉
    〈Document〉
      〈doc_id〉0〈/doc_id〉//系统内部的 doc_id 编号，顺序递增
〈title〉E:\Projects\LJSearch\LJSearch\TestDLL\Corpus1021\Corpus\Meteorological Bulletin Automatic Generation based onSpatio-Temporal Reasoning. txt〈/title〉//说明：字段名 title 由字段类型定义定制需要检索显示出的字段
        〈content〉Meteorological bulletin has more and more diversified, large scale, highly integrated requirements and potential demands from whole society…〈/content〉//说明：内容摘要，关键词红显
      〈/Document〉
〈/LJSearch-Result〉
```

8.4.3 JZSearch 检索语法说明

JZSearch 的检索语法类似于 SQL 语句，由下面格式的语句组成。

{[FIELD] fieldname/ * [搜索运算符] value1 value2 ... value_n} * | [SORT] sort_type

说明如下：

(1) [FIELD]为搜索系统保留字。

(2) Fieldname 为系统设置的字段名称，该字段必须在 3.2 JZSearch_FieldAdd 中定义过，否则系统不接受查询请求；其含义为在该字段上进行搜索运算。

(3) Fieldname 也可以为“*”，含义为通配符搜索，不需要指定特定字段，所有涵盖在通配搜索范围的字段都纳入搜索的范畴。

(4) AND/OR/NOT/NEAR/RANG/MIN/MAX/PREF/PREC/FUZZY，具体含义见后面的详细描述。

(5) Value1 Value2 等为参数列表。

(6) 大括号的语句可以反复叠加使用，各个语句之间是逻辑与的关系。

(7) 各个部分之间参与空格或者〈TAB〉键分开，无论是保留字还是参数均不区分大小写。

(8) [SORT]sort_type 用于设定当前搜索结果的排序方式。

1. 搜索与运算符 AND

表示后续参数列表的逻辑与关系，举例说明如下。

[FIELD] title [AND] 基金 公司

含义：搜索 title 字段中同时包含“基金”与“公司”的文档记录。

2. 搜索或运算符 OR

含义是后续参数列表的逻辑或关系，举例说明如下。

[FIELD] title [OR] 基金 公司

含义：搜索 title 字段中包含“基金”或“公司”的文档记录。

[FIELD] * [OR] 基金 公司

含义：搜索任意字段中包含“基金”或“公司”的文档记录。

3. 搜索非运算符 NOT

含义是后续参数列表的逻辑非关系，举例说明如下。

[FIELD] *[AND] 基金　[FIELD] title [NOT] 公司 中心

含义：搜索任意字段中包含“基金”，但是标题中不包含“公司”也不包含“中心”的文档记录。

4. 搜索邻近运算符 NEAR

格式为：[FIELD] fieldname/ * [NEAR] value1 value2 value3

后续参数为 value1、value2、value3；前两个参数 value1、value2 为搜索内容，第三个参数 value3 为邻近的距离值；只有两个参数时，默认第三个参数 value3

为 10。

上述语句含义是搜索内容 value1、value2 必须同时存在，而且距离不能超过 value3 个词，类似 AND，但比 AND 严格。

[FIELD] * [NEAR]汪精卫 卖国贼 9

含义：搜索任意字段中同时包含“汪精卫”和“卖国贼”的文档，同时要求两者的距离不能超过 9 个词，采用该语法信息过滤将更有针对性。

5. 搜索智能模糊运算符 FUZZY

含义是搜索方式为模糊匹配，尽可能地匹配搜索串，但模糊处理，不做严格的匹配，举例说明如下。

[FIELD] * [FUZZY] 张华平 kevinzhang 精准搜索内核

含义：搜索任意字段中含有“张华平 kevinzhang 精准搜索内核”的文档，按照相关度自动排序。

6. 搜索范围运算符 RANG

格式为：[FIELD] fieldname/ * [RANG] value1 value2。

后续参数为 value1、value2。

含义：字段 fieldname 的值必须在 value1、value2 之间的范围内；fieldname 字段必须为数值型字段；否则，非法。

举例说明如下。

[FIELD] price [RANG] 1.0 9.0 [FIELD] name [AND] 牛奶 儿童

含义：搜索价格在 1.0～9.0，且商品名包含“儿童”、“牛奶”的记录

7. 搜索最小值运算符 MIN

格式为：[FIELD] fieldname/ * [MIN] value1。

后续参数为 1value1。

含义：字段 fieldname 的值必须大于或等于 value1；fieldname 字段必须为数值型字段。

举例说明如下。

[FIELD] price [MIN] 1.0 [FIELD] name [AND] 牛奶 儿童

含义：搜索价格不低于 1.0，且商品名包含“儿童”、“牛奶”的记录

8. 搜索最大值运算符 MAX

格式为：[FIELD] fieldname/ * [MAX] value1。

后续参数为 value1。

含义：字段 fieldname 的值必须不大于 value1；fieldname 字段必须为数值型字段。

举例说明如下。

[FIELD] price [MAX] 9.0 [FIELD] name [AND] 牛奶 儿童

含义：搜索价格不高于 9.0，且商品名包含“儿童”、“牛奶”的记录。

9. 搜索前缀运算符 PREFIX

格式为：[FIELD] fieldname/ * [PREF] value1。

后续参数为 value1

含义：字段 fieldname 的内容必须以 value1 开头；fieldname 字段必须为字符型字段。

举例说明如下：

[FIELD] name [PREF] 张

含义：姓名字段 name 必须以“张”作为前缀开头。

10. 搜索精准运算符 PRECISION

格式为：[FIELD] fieldname/ * [PREC] value1。

后续参数为 value1。

含义：字段 fieldname 的内容必须精准匹配 value1；fieldname 字段必须为字符型字段。

举例说明如下：

[FIELD] name [PREC]abc

含义：姓名字段 name 必须以“abc”精准匹配，如“abcd”不作为匹配结果。

[FIELD] id [PREC] 123

含义：姓名字段 id 必须以“123”精准匹配，如“1234”或者“0123”均不作为匹配结果。

11. 复合运算符 COMPLEX

格式为：[FIELD] fieldname/ * [COMPLEX] keyword1|| keyword2||…|| keywordN word1|| word2||…||wordM dist。

后续参数至少 2 组：

keyword1|| keyword2||…||keywordN：关键词列表（中间用||隔开）

word1|| word2||…||wordM：关键词列表（中间用||隔开），关键词列表至少需要两组，可以多组兼容。

dist：表示距离，不设置则不考虑之间的距离。

表示的含义：

{keyword1 OR keyword2 OR…OR keywordN} AND {word1 OR word2 OR …OR wordM}

含义：如有距离，则表示集合{keyword1 OR keyword2 OR…OR keywordN}与集合{word1 OR word2 OR…OR wordM}，出现的词必须在 dist 范围以内。

举例说明如下：

[field] title [complex] 台湾||解放军||孕妇　甲流||人事

含义：要求 title 字段内，同时出现{台湾||解放军||孕妇}{甲流||人事}。

[field]content [complex] 统计局||中国统计局||CPI　数据||专家 10

含义：要求 content 字段内，同时出现{统计局||中国统计局||CPI }{数据||专家}，且两者之间的距离必须在 10 个词内。

12. 列出所有索引的结果内容 LISTALL

格式为：[CMD] LISTALL。

含义：不给定搜索条件，要求列出所有的索引内容。

举例说明如下：

[cmd] listall

含义：列出所有的索引内容，默认按照 docid 从小到大排序。

[cmd] listall [sort] docid_reverse

含义：列出所有的索引内容，按照 docid 从大到小逆序。

13. 正负面情感搜索

正负面搜索采用特殊的搜索词，可以用前面所述的任何一种操作符号。特殊的搜索词如表 8.6 所示。

表 8.6　JZSearch 正负面搜索

类型名	说明
##负面 JZSearch##	搜索负面信息
##正面 JZSearch##	搜索正面信息

举例说明如下：

[field] content [near] 张华平##负面 JZSearch## 12

含义：在 content 字段搜索“张华平”和负面信息相隔 12 个词范围内的文档。

[field] content [and] 张华平##正面 JZSearch## [sort] publishtime desc

含义：在 content 字段搜索 包含“张华平”和正面信息的文档，且按照发表时

间倒序排列。

14. 删除显示运算符 Deleted

格式为：[deleted] on/off。

含义：搜索结果中包含删除文档的部分是否需要显示出来，如果为 on 将显示；否则，不显示。

举例说明如下：

[FIELD] title [AND] 基金 公司 [deleted] on

含义：将所有基金公司的信息找出来，如果删除文档中有相关的内容也要显示出来。

8.5　JZSearch 搜索引擎管理

8.5.1　搜索引擎可视化管理客户端

JZSearch 提供了可视化的管理客户端，界面如图 8.3 所示。

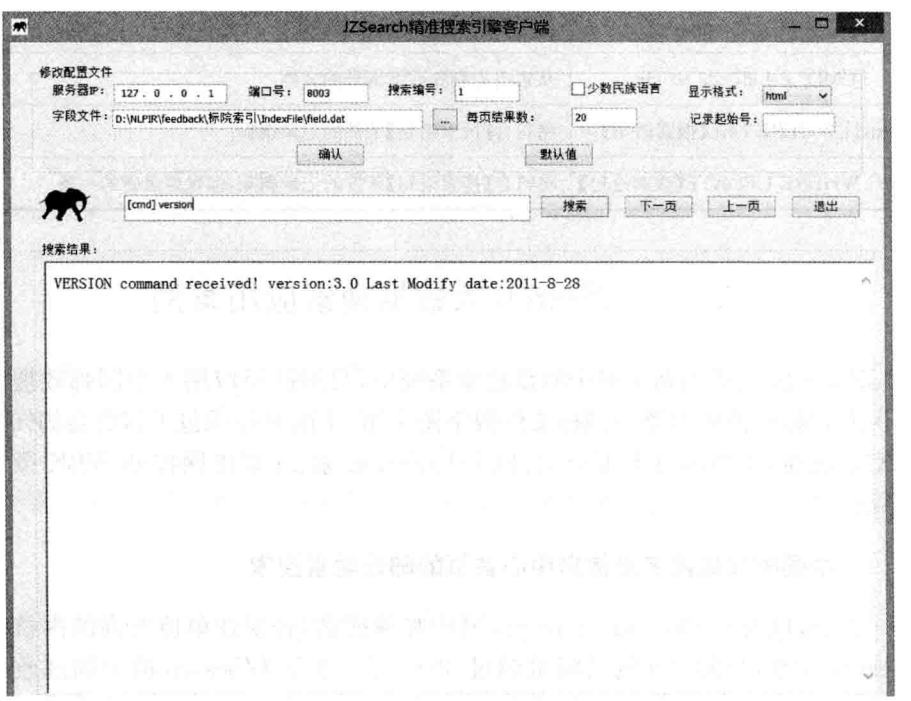


图 8.3　JZSearch 可视化管理客户端

通过管理客户端，可以与搜索引擎实时交互，搜索提交的查询语句，也可以发送管理命令。

8.5.2 客户端管理命令语法

客户端的命令都是采用搜索服务控制命令实现的，服务端控制命令类似于搜索语法。具体细节如表 8.7 所示。

表 8.7 JZSearch 管理命令

命令格式	说明
[CMD] RELOAD	重新加载新增的索引内容，系统会自动维护，不建议使用
[CMD] MERGE	对索引数据进行优化归并，系统会自动维护，不建议使用
[CMD] QUIT	将系统搜索服务停止
[CMD] Version	查询搜索引擎的版本号
[CMD] Backup	索引自动备份，系统会自动维护，不建议使用
[CMD] Restore	索引自动恢复，系统会自动维护，不建议使用
[CMD] del id1 id2 … idn	从索引中删除后续编号的文档
[cmd] WHERE-DEL【搜索语句】	将符合【搜索语句】条件的记录删除
[cmd] WHERE-UPDATE【搜索语句】	将符合【搜索语句】条件的记录删除，但保留最新的一条

8.6 JZSearch 大数据搜索应用案例

JZSearch 已应用到了多个海量搜索系统中，目前已经应用于中国邮政搜索引擎、河北省标准搜索引擎与国标文件数字图书馆、中国对外承包工程商会的知识搜索、富基融通（纳斯达克上市公司：EFUT）商品搜索、北京市网控办 WBK 微博监控系统。

8.6.1 中国邮政集团名址信息中心首页的邮址垂直搜索

访问地址为 www. cpdc. com. cn，其中邮编搜索、企事业单位查询的内核均由 JZSearch 实现，目前每年的访问量超过 3000 万。基于 JZSearch 的中国邮政搜索引擎效果如图 8.4 所示。

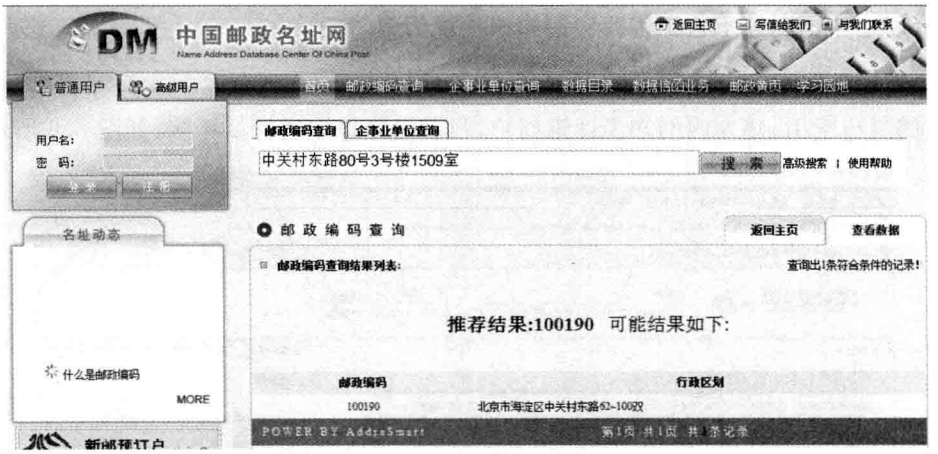


图 8.4　基于 JZSearch 的中国邮址搜索引擎效果

8.6.2　河北省标准化研究院的标准搜索

该系统的搜索部分均由 JZSearch 对其国内外标准数据库及全文文件库进行索引，提供全省上万家企事业单位与政府机构搜索，搜索段为 C/S 模式。基于 JZSearch 的标准文献垂直搜索效果如图 8.5 所示。

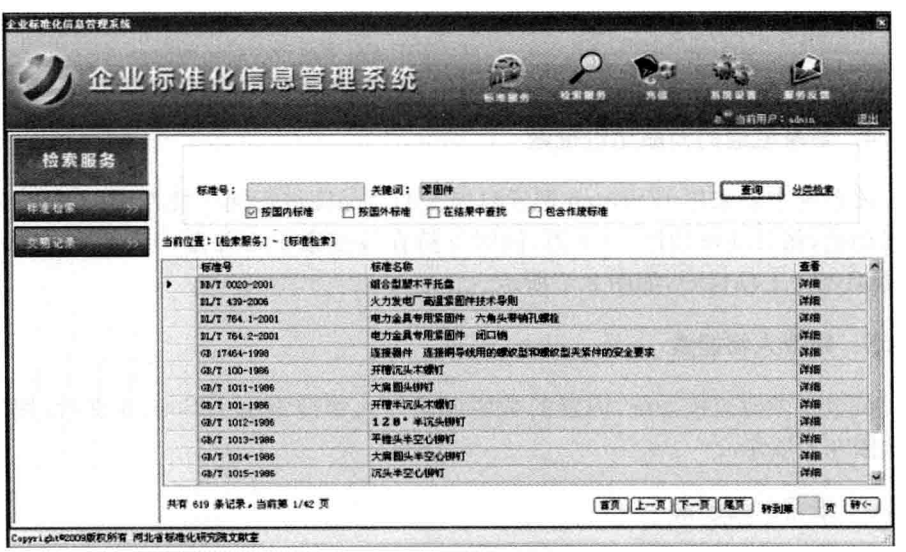


图 8.5　基于 JZSearch 的标准文献垂直搜索效果

8.6.3 中国对外承包工程商会的知识搜索门户

该系统主要采用 JZSearch 搜索对中国对外承包商会内部 100 台办公电脑的文档进行索引，建立内网知识搜索门户，目前已经稳定运行 2 年多，如图 8.6 所示。



图 8.6 基于 JZSearch 的中国对外承包工程商会的知识搜索

8.6.4 富基融通的商品比价搜索

该系统主要采用 JZSearch 搜索对庞大的商品库进行同一性判别，并提供比价搜索功能，每日新增数据 1000 万，同时支持在线搜索 1 万并发。该系统已经内嵌到其 MyStore 项目中，如图 8.7 所示。

8.6.5 微博人物搜索

黄金眼微博监控系统，内置的微博搜索与人物搜索由 JZSearch 支持，搜索效果如图 8.8 所示。

8.6.6 维吾尔语搜索

利用 JZSearch，将中英文词典支持换成少数民族语言解析器，可以支持大多数民族语言的搜索。维文搜索效果如图 8.9 所示。



图 8.7　基于 JZSearch 的富基融通商品比价搜索系统

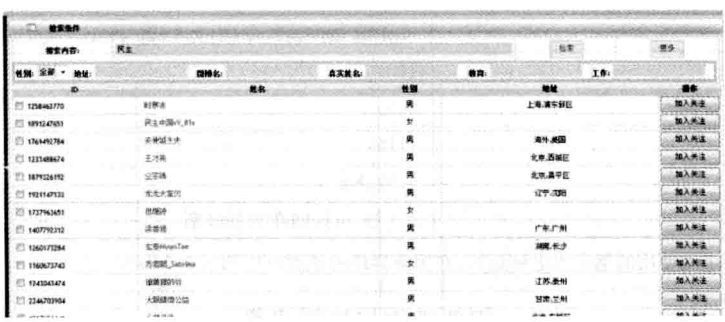


图 8.8　基于 JZSearch 的微博人物搜索系统



بانى بەت قولۇڭبەت سەلاھىتىدەرىر خەت ساندۇقىدالاقىلىرشىڭباش بەتمەركىز文كەس人民网维吾尔--خەلق نورى
خەۋەرلىرى خەلقدا؛ خەۋەرلەرشى نجالۇ #x0D; #x0D; #x0A; روھى مەسلىكەت
خەۋەرلىرى نى قىتى سانداجمىنى بەشىنارى نىتەتەر بىيەسى لەت ۋە مەدەنىيەتنۇ ھىيت ئاسرالىدى نغىزىرىق نۇقتاشەخسلەر سورىتلىك
خەۋەرلەرمانىرى نىلمەخسۇس سەھىيەدەھەر ۋە نجىدىزىيگى خەۋەرلەر مەسلىكەتلىرى نجلەر ۋاسكىتبول كەسىپى بىرلەشمە
مۇسابىقەسى نى ئالۇ نوموزىمى ھەل قىلغۇچ مۇسابىقەسى نده گۇلگۇلۇك كوساندەسى شى نجالۇ كوساندەسى نى بەگدى شى نجالۇ
تورپاندا كەم كورولى دى غان بوراتلىق ھۆا رايى كورولدى گەنسۇ جاڭبيا قۇم بوراتلىق ھۆا رايى كورولدى 1-يەسى لەدە سىزىتتىزىن
شى نجالۇ سەلىنغان مەلۇمۇغ نوخشاش ۋاقىتتىكى-ئىن 4 بيرىم ھەسسە ئاشتى روسى يەنى ئالۇ نوتجى تورركومەسى كى بارەدم
بۇيۇملىرى بيتىپ كەلدى شى نجالۇ قۇمۇلدا قۇم بوراتلىق ھۆا رايى كورولدى نالدىن مەلۇمات: نادىل ھوشۇر خەلق نورى دا
شە نجالۇ خەلق نۇقۇ ۋى سا ... سەھىيەت ئەسكە ئەلۇ تەھقىقات ... ھەتتا ئەلۇ ...

图 8.9　基于 JZSearch 的维吾尔语搜索系统

8.7 本章小结

本章主要介绍了 JZSearch 大数据搜索引擎的关键技术，探讨了如下内容。

(1) 介绍了 JZSearch 的基本属性特征。

(2) 研究了 JZSearch 搜索引擎的体系架构，并介绍了在线增量索引的状态转换图的设计。JZSearch 主要包括索引器、搜索器、词法分析器，以及配套的管理器与数据库适配器，并提供了可供二次研发的 SaaS API。

(3) 依次介绍了 JZSearch 索引技术、搜索技术，以及管理工具等。

JZSearch 大数据精准搜索引擎的性能测试结果如表 8.8 所示。

表 8.8 JZSearch 性能测试

项目	参数
测试机器配置	ThinkPad x220i 内存 2G，Intel Core i3－2310M CPU @ 2.10GHz 2.10GHZ
测试数据	NLPIR 新闻语料库(中文 3 686 479KB 英文 2 249 607KB)
使用内存	1GB
占用 CPU	25%
索引时间	1115s
索引速度	5,323KB/s
搜索速度	＜1 秒 可控制在毫秒级别

注：测试在个人使用的笔记本电脑运行，在服务器端的速度可以超过 10MB/s。

相关工作与扩展阅读

检索模型(张华平，2010)就是一种算法，该算法的处理对象是查询 Q 和文档集合$\{D_1, D_2, \cdots, D_n\}$，处理过程就是计算每篇文档 $D_i(1 \leqslant i \leqslant n)$和这个查询的相似度 $SC(Q, D_i)$。

常用的检索模型有如下几种。

(1) 向量空间模型，将查询和文档都表示为词项空间中的向量，进而可以计算这两个向量之间的相似度。

(2) 概率模型，基于文档集中每个词项在相关文档中出现的可能性计算出概率。对文档和查询匹配的所有词项计算联合概率，从而得出文档与查询的相似度。

(3) 语言模型，针对每一篇文档，建立一个语言模型，并且计算出各文档“生成”查询的概率。

(4) 推理网络，采用贝叶斯网络来推断文档和查询间的相关性。推理过程基于文档中的“证据”，这种“证据”是文档中可以作出文档相关性推断的依据。推理的可信度即为文档查询的相似度。

(5) 布尔检索，在原始的布尔查询结果中，对每篇文档都赋予一个得分，这样就可以进行查询结果排序。具体的做法是将每个查询词项赋予一个权重，进而使用这个权重计算文档与查询的相似度。

(6) 隐性语义检索(LSI)，使用词项-文档矩阵来表示文档集中出现的词项。这个矩阵通过奇异值分解(singular value decomposition，SVD)来降维，奇异值分解可以过滤掉文档中的噪声，这样两篇有相同语义的文档在多维空间中距离会比较接近。

(7) 神经网络方法，网络中包含一系列神经元(或者称作节点)，这些神经元在激活查询到文档的触发链接的过程中进行传导。网络中每个链接的权重被传到文档，这些权重汇集起来作为查询和文档之间的相似度。人们往往会预先定义好一批相关文档集和不相关文档集，然后再调整神经网络链接上的权重，以达到“训练”神经网络的目的。

(8) 遗传算法，可以通过进化的方法来得到查找相关文档的最优查询过程。初始的查询可以使用随机的权重，也可以使用估计值。通过修改这些权重来产生新的查询。接近已知相关文档的新查询得以幸存，而适应度较小的查询在随后的过程中被淘汰。

(9) 模糊集检索，每篇文档被映射到一个模糊集，这种集合不仅包含元素，还包含与每个元素相关的权重，用来表示该元素的隶属度(相似度)。该模型将布尔查询映射为模糊集的相交、合并和补集等操作，这样就可以计算出每篇文档与查询相关的隶属度。

哪种模型是最好的仍然是备受争议的话题，这有待于进一步的研究。尽管 TREC(TREC 2013)(文本检索会议)的任务是评价完整的信息检索系统，但是一个系统中除了包含检索模型和一些实用策略，还包括一些其他的实现细节。到目前为止，还没有一个可以明确地在给定的统一的系统框架上进行检索模型间比较的结论。

相关的开源搜索系统有：Apache 组织的 Lucene(采用 Java 开发)(Lucene，2013)、俄罗斯的 Sphinx(采用 C/C++开发)(Sphinx，2013)、我国的 FirteX(采用 C/C++开发)。

参 考 文 献

张华平，李恒训，刘治华．2010. 信息检索算法与探索法．北京：人民邮电出版社．

Bell TC，Cleary J G，Witten I H. 1990. Text Compression. New York：Prentice Hall.

FirteX. 2013. http：//sourceforge. net/projects/firtex/.

Grossman D. 1995. Integrating Structured data and text：a relational approach. PhD thesis，George Mason University.

Lucene. 2013. http：// lucene. apache. org/.

Sphinx. 2013. http：//sphinxsearch. com/.

TREC. 2013. http：// trec. nist. gov.

Witten I，Moffat A，Bell T. 1999. Managing Gigabytes. New York：Van Nostrand Reinhold.

第 9 章　面向大数据的句子检索与新颖性监测

9.1　概　　述

在针对大数据的搜索中，普通的检索技术能在一定程度上满足对大数据的搜索需求，但结果可能会包含大量无关、冗余的信息，信息粒度偏大。如果用户能够提炼自己的需求，并以适当的关键词表达出来，检索效果可能会更好。但这对没有检索经验的普通用户来说有一定难度。另外，在大数据的搜索中，用户常常需要粒度较小、相关而又新颖的信息。为此，本章给出比文档粒度更小的句子级的搜索与新信息检测算法，进一步排除冗余，并检测出新信息。

句子级新信息检索涉及信息检索、信息过滤、文本分类、自动文摘等相关技术，它是在信息检索基础上的细化。新信息检测具有时序性，即对于内容相同或相近的句子，最新出现的应被检出(先前出现的相似内容属过时信息)。句子检索的一般步骤是依据检索算法计算出句子与正向查询的相关度，同时兼顾逆向查询的相关度，并选择正向查询相关度超过一定阈值、逆向查询相关度较小的句子作为检索结果返回。表 9.1 从信息粒度、相关性检索、新信息过滤、查询表达与技术难度等方面比较了句子级新信息检测与传统技术间的差别。

表 9.1　句子级新信息检测与传统技术比较

技术	信息粒度	相关性检索	新信息过滤	查询表达	技术难度
信息检索	文档、网页	相关	没有排重处理 存在大量冗余	关 键 词	基本能实用化
自动问答	词语、短语	精确相关	唯一答案，无冗余	自然语言	仍不成熟
句子级 新信息检测	句子	相关	按报道时序进行 新信息检测， 基本无冗余	自然语言	正在探索， 技术相对成熟

本章阐述句子检索的查询扩展方法和新颖性检测算法，并给出在 TREC 数据集上进行的实验结论。依据本章技术方法研制的 Noovel 原型系统取得了较好的性能。在句子检索方面，原型系统 Noovel 采取向量空间模型、概率检索模型与语言模型。为了克服句子的局限性，引入查询扩展的技术，包括借助于 WordNet 的语义衍生扩展、伪相关反馈扩展、采用高频共现词语的局部共现扩展。TREC2003 数据集上的实验表明，在浅层语言分析的基础上，简单向量空间模型可以达到最好

的结果，受到语义资源和分析深度的限制，当前阶段的语义扩展作用有限，而伪相关反馈与局部共现扩展都能够帮助提高句子检索的性能，可见局部共现扩展是很有潜力的查询扩展与文档扩展的技术。

9.2　句子检索的查询扩展方法

普通的文档往往包含比句子多数十倍的实词，一般可通过分析高频出现的词语来强调其内在含义，同时这也增加了其被检索到的可能，但单句中具有实际意义的词语往往只有 4～5 个，且任何词语几乎都不可能重复出现。因此，仅依靠简单的向量空间模型，句子检索往往很难达到文档检索的效果。这里通过下面两组例子来说明简单的向量空间模型方法在句子检索中存在的局限性。

第 1 组句子。

(1a) Daily we read news stories about dissatisfaction with managed care, Medicare fraud and overbilling.
(1b) Eighty percent agreed with this.

第 2 组句子。

(2a) However, the Scottish team was the first to make a clone from adult animal cell.
(2b) The seminar was held in the context of a recently reported sheep cloning case in Britain.

第 1 组和第 2 组的句子均来自于同一个主题的句子。第 1 组分别标明对医疗改革的两种对立态度，即dissatisfaction(不满)与agreed(支持)。第 2 组讨论的都是克隆羊的问题，(2b)中的sheep(绵羊)属于animal(动物)的一个种类。从语义上来说，两者存在着很强的相关性。然而仅仅根据词形计算，第 1 组的相关性为 0，第 2 组的两个句子中仅有clone 相同。可见，向量空间模型相关度计算的结果往往不能正确反映它们之间的关系。虽然相关词语不能匹配的问题在文档检索中也存在，但是在句子检索中，这个问题要严重得多。为此，需针对句子特点，对主题查询做进一步扩展。在句子检索阶段，主要采用三种查询扩展办法，即语义扩展、伪相关反馈扩展与局部共现扩展。

9.2.1　语义扩展

针对上面提到的问题，一个直接的解决方法是引入语义体系，抽取出词形不同的相关词语之间共有的语义特性。例如，上面例子中的“dissatisfaction”与“agreed”同属于情绪和态度范畴，而“sheep”是“animal”的下位概念，因此可引入在

线英文词汇数据库 WordNet。WordNet 用一组同义词的集合 Synset 来表示一个概念，每一个概念有一段描述性说明，有多个语义的词分别对应不同的 Synset。Synset 之间还存在上下位关系、同义反义关系、部分整体关系等。我们主要考虑名词上下位及同义关系，并兼顾形容词近义与反义关系，如图 9.1 和图 9.2 所示。

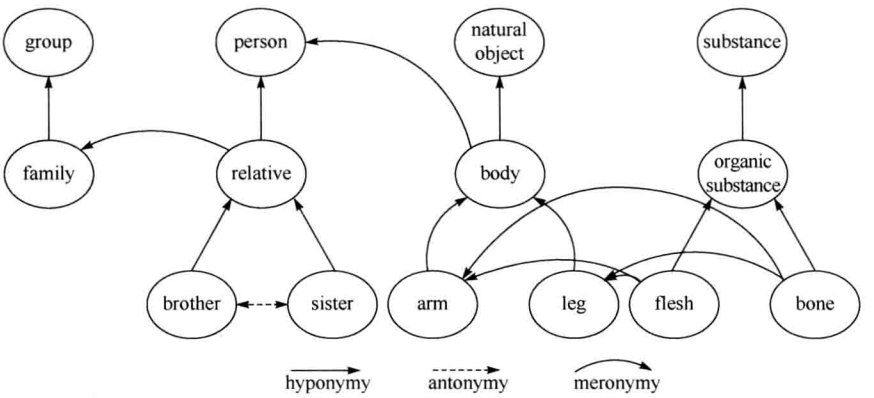


图 9.1 WordNet 名词的上位、下位与反义等语义关系

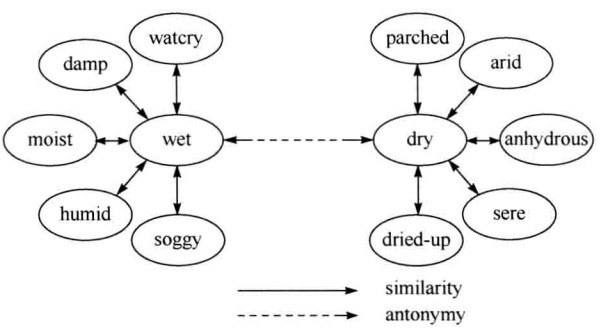


图 9.2 WordNet 中形容词的近义、反义等语义关系

借助 WordNet 的语义体系，本章提出两种语义扩展方法，即语义映射与词语衍生。

1. 语义映射

语义映射是将词语组成的向量空间，通过语义关系转换映射到概念空间之上，原来的词语向量转换为概念向量。具体做法是将词语的各个语义按一定的加权策略映射到概念空间，同时将概念空间上的同义概念、上下位概念、近义概念等进行归并，基本思想如图 9.3 所示，最后在概念空间上进行向量运算并检索

结果。

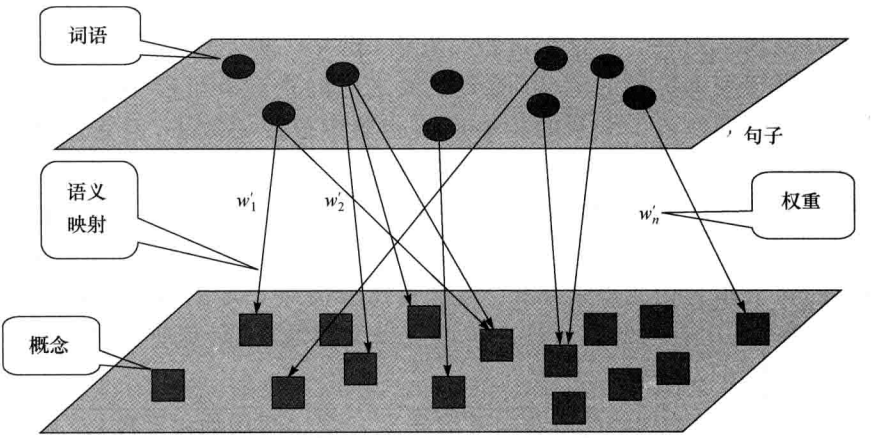


图 9.3　语义映射与扩展示意图

语义映射的目的在于将词语转化为语义上更一般性的概念，避免同义相关词语词形匹配不一致的问题，可扩展句子查询相关度的计算，但语义映射也存在缺陷。

(1) 它过分依赖一份完备的语义体系，实际上目前的语义体系往往难以达到这种完备高度。

(2) 语义概念之间的关系往往千丝万缕，往往很难满足向量空间模型中的独立性假设。

(3) 词语到概念的映射存在歧义且计算代价高。

2. 词语衍生

词语衍生是给定某个词语 w，通过 WordNet 得到 w 对应的所有概念(Synset 集合)，将所有属于同一 Synset 或上下位 Synset 中的词语组成一个扩展词集合 S_{ext}，假设所在主题的特征词集合为 T_{feature}，则可将词语 w 衍生为词语集合 DS_w，即

$$\text{DS}_w=\{w' \mid w' \in S_{\text{ext}} \cap T_{\text{feature}}, w' \neq w \wedge \text{sf}(w') \geqslant 2\}$$

其中，sf(w')表示包含 w'的句子数目，即 w'衍生出的词语是那些在当前主题的句子集内出现过 2 次以上的同义词或其上下位词。

接下来，将向量中的 w'扩展为(w', DS_w)。DS_w 相应的权重由对应的 Synset 分布概率结合 w'的特征权重估计而得，最终的向量需要进行二次规格化处理。通过这种方式，就可简单地解决词语不匹配的问题。

9.2.2　伪相关反馈扩展

早期的相关反馈是通过人工进行的。用户提交原始查询后逐个浏览查询

结果并给出相关性判定；系统根据相关性判定，将查询中的相关信息加强，非相关信息减弱，然后将修改后的查询再次提交检索系统。如此反复进行可以得到较好的检索效果。从这个过程可看出，相关反馈必然包括对查询的修改，但不一定属于查询扩展的范畴，因为反馈之后的查询有可能不包括初始查询中的关键词。即使人工干预的相关反馈能取得很好的效果，但它仍是一种不实用技术，因为实际的检索过程一般都是一次性的，用户并不愿意也不会提供更多的反馈。因此，研究者们引入了伪相关反馈技术，不需人工进行相关性判定，从而提高了自动化程度。伪相关反馈的基本思想是自动地选取查询相关度较高的句子作为相关检索结果，将这些句子按照一定的加权比例对查询进行正向反馈，采用它们的向量进一步修正查询向量，从而实现查询扩展。具体算法流程如图 9.4 所示。

输入：查询 Q、相关文档中的所有句子集合 S、反馈比例 θ、加权系数 λ。

1. 采用简单的向量空间模型，计算出查询 Q 与相关文档中的所有句子集合 S 的查询相关度。

2. 按照相关度从大到小的顺序，对句子集合 S 重新排序。

3. 选取 S 的前 $|S| * \theta$ 个句子向量，作为正向伪相关反馈向量集合 S_{pseudo}。

4. 按照下面的公式修正查询 Q 的向量，即

$$\vec{Q} = (1-\lambda)\vec{Q} + \lambda \sum_{\vec{s} \in S_{\text{pseudo}}} \vec{s}$$

图 9.4　句子查询的伪相关反馈扩展算法

句子检索可能遇到的困难在于检索到那些并不包含查询词的句子，因为伪相关反馈只考虑正向伪反馈，而没有加入逆向伪反馈。

9.2.3　局部共现扩展

局部范围内频繁共同出现的词对，在多数情况下存在某种内在的关联性，因此可以利用高频共现的词对实现对查询向量与句子向量的进一步扩展。设词对为 $<ID_1, ID_2>$，在新信息检测中将局部范围窗口设置为一个单句内部，选取一个句子中共现的高频词 ID_2 来扩展 ID_1 的条件是 $SF(ID_1, ID_2) > \theta_1$ 且 $P(ID_2 | ID_1) > \theta_2$。其中，$SF(ID_1, ID_2)$ 为同时出现 ID_1 与 ID_2 的句子数目，$P(ID_2 | ID_1)$ 为共现条件概率，θ_1 与 θ_2 为阈值，可用于调节扩展词的质量与数量。这里以 TREC 的主题 N1 为例（这是一个讨论“partial birth abortion ban”反堕胎方面的评论型主题），从 25 篇相关文档的浅层分析结果中统计了所有词对的共现概率，最终可得到如表 9.2所示的高频局部共现词对表。

表 9.2　TREC 主题 N1 中的高频局部共现词对表

词语 ID_1(词语$_1$)	词语 ID_2(词语$_2$)	共现频率 $SF(ID_1, ID_2)$	共现概率 $P(ID_2 \mid ID_1)$
52 104 (partial)	7333 (birth)	26	0.619048
52 104 (partial)	5501 (ban)	19	0.452381
7333 (birth)	52 104 (partial)	26	0.433333
52 104 (partial)	482 (abortion)	17	0.404762
7333 (birth)	5501 (ban)	18	0.3
7333 (birth)	482 (abortion)	16	0.266667
5501 (ban)	482 (abortion)	24	0.186047
52 104 (partial)	15 879 (court)	7	0.166667
5501 (ban)	52 104 (partial)	19	0.147287
5501 (ban)	7333 (birth)	18	0.139535
52 104 (partial)	56 001 (procedure)	5	0.119048
52 104 (partial)	69 802 (supreme)	5	0.119048
7333 (birth)	15 879 (court)	7	0.116667
69 802 (supreme)	15 879 (court)	6	0.107143
69 802 (supreme)	7333 (birth)	5	0.089286
69 802 (supreme)	52 104 (partial)	5	0.089286
7333 (birth)	56 001 (procedure)	5	0.083333
7333 (birth)	69 802 (supreme)	5	0.083333
69 802 (supreme)	5501 (ban)	4	0.071429
482 (abortion)	5501 (ban)	24	0.066667

通过上表可以发现很多有意思的现象，例如堕胎与最高法院(supreme)、法院(court)、法律程序(procedure)的内在关联性。尽管它们具有完全不一样的语义，但从内在主题相关性来说，它们又具备潜在关联性，这是复杂的语义分析都很难解决的现象。需要指出的是，局部共现扩展和具体统计采样的文档集合密切相关。在不同立场色彩的文档集合上统计高频共现词对，最终的词表很可能大不相同，扩展的结果也会受到影响，它们会更多地代表采样文档集的特性。直观地说，如果在非相关文档集合上进行采样统计分析，那么形成的高频共现词对表会更多地偏向于非相关文档，最终干扰信息的正确检索。局部共现扩展存在被滥用甚至错用的可能，因此需要有选择性地选取采样文档集合。具体实现时，可利用伪相关反馈思想，选取查询相关度排名靠前的句子集合，并在此基础上进行局部共现统计，利用高频的局部共现词对表扩展句子向量中的词语。假定句子的向量为 s，其中有一个特征词语为 t，其权重为 w。t' 是 t 的高频局部共现词，那么可以将 t' 加入到 s

中，其权重为 $w * P(t'|t)$，而 t 的权重相应修改为 $w * (1-P(t'|t))$，其他的高频共现词依此类推，最终实现句子检索的扩展。

9.3 语言模型检索

9.3.1 概述

语言模型是一种基于概率的检索模型，它认为每个文档对应一个统计语言模型(称为文档语言模型)，描述该文档中各个单词的统计分布特征，而每个文档看作是由其语言模型抽样产生的一个样本。查询 q 也可看做是由该文档的语言模型抽样产生的一个样本，因此可根据每个文档语言模型抽样生成查询的概率来排序，概率值越大，则该文档就越满足查询要求，作为检索结果的可能性就越大。

传统概率检索模型将文档 d 与查询 q 的相关度排序函数定义为事件 R(文档是否满足检索要求)的概率，即 $f(q,d)=P(R|d, q)$。相关度排序函数定义虽较直观，但相关性是一个抽象的概念，也无法给出 R 的定义，所以该模型在理论上存在很大的模糊性。基于语言模型的检索相关度排序函数则定义为由文档 d 的语言模型生成查询 q 的概率，即 $f(q,d) = P(q|d)$。它建立在统计语言模型理论基础上，为便于操作，可以在文档词语概率分布基础上直接计算出相关度排序函数。与向量空间模型、概率检索模型等类似，这里也假设查询词之间是相互独立的关系，即一元语言模型，因此有

$$P(q \mid d) = \prod_i P(q_i|d)$$

所以，最后将 $P(q|d)$的求解转化为如何估计概率 $P(q_i|d)$。

统计方法的核心是利用已知事件的发生频率来估计该事件客观存在的概率，需要根据已知事件的发生频率来尽可能地“逼近”其内在的发生概率。某个事件概率是客观存在的，观测到的频率符合一定规律，但是具有很大的随机性。另外受客观条件限制，人们也不可能观测到所有现象。我们估计出来的概率往往和实际的概率存在一定的差距，在样本数量有限的情况下，简单最大期望估计甚至可能完全背离客观实际。例如，抛硬币 2 次，很可能 2 次都是正面，仅仅根据观测到的小规模结果，我们会得到抛硬币得到正面的概率为 100%，而反面的概率为 0。这显然是不合理的。

为了解决因数据稀疏引起的零概率问题，需要采用数据平滑的技术进行处理，其基本思想是从已发生事件的概率中扣除一部分，然后把这些“多余”的概率重新分配给那些未发生的事件和已发生的低频事件中。数据平滑的好坏直接影响到语言模型的性能。文献(Chen et al.，1996)对数据平滑方法进行了很好的综述。一般来说，如果 q_i 出现在文档中，可以按照它的词频信息计算 $P(q_i|d)$；如果 q_i 不出现，在没有其他先验知识的情况下，可按照 q_i 在所有文档集合中出现的情况来估

计它的概率。文献(Zhai,2001)得到的 $P(q|d)$表达式为

$$\log P(q \mid d) = \sum_{i} \ln P(q_i \mid d)$$

$$\log P(q \mid d) = \sum_{\mathrm{TF}_{d,qi}>0} \ln P(q_i \mid d) + \alpha_d \sum_{\mathrm{TF}_{d,qi}=0} \ln P(q_i \mid C)$$

其中,C 为当前主题的所有文档集合。

如果 q_i 在 C 上仍然没有出现,还可进一步引入大的真实语料库来估计 q_i 在整个通用语言环境的概率。

9.3.2　句子级语言模型及其改进

句子检索中,语言模型需要进一步平滑处理。在文档检索基础上,采用下面公式来计算句子 s 的查询相关度,即

$$\mathrm{Sim}(q,s) = \ln P(q \mid s) = \sum_{q_i \in q} \ln[P(q_i \mid s) + \alpha_s P(q_i \mid d) ++ \alpha_d P(q_i \mid C)]$$

其中,d 为当前句子所在文档;C 为主题所有的文档集合,目的在于将句子检索与文档检索进行结合,利用所在文档来平滑句子 s 生成查询 q 的概率。

9.4　句子检索实验与分析

在 TREC2002 与 TREC2003 新信息检测任务的数据集基础上,我们设计了三组实验。

(1) 浅层语言分析的贡献度实验,主要是通过实验来评估浅层语言分析对新信息检测的潜在贡献度。

(2) 三种句子检索模型的基准实验,主要是测试向量空间模型、概率检索模型与语言模型的基本性能。

(3) 查询扩展实验,在简单向量空间模型的基础上,测试各种查询扩展技术的对比性能。

需要说明的是,TREC2002 与 TREC2003 给定的文档都是与主题相关的,也就是说,可以忽略文档检索过程,排除不相关文档的干扰。

9.4.1　浅层语言分析的贡献度

新信息检测是一个复杂的问题,涵盖浅层语言分析、文档检索、句子检索、新信息检测等四个过程,且每个环节都存在多种变化因素。各种各样的因素都会直接或者间接地影响新信息检测的最终性能,甚至说某个技术环节的小技巧都很可能提高或者降低最终性能。为了评估浅层语言分析的作用,我们尽可能地减少其他因素的干扰。因此,只考虑句子检索的性能,因为句子检索结果直接决定后期的新

信息检测，从句子新信息检测的角度出发，最大的影响因素是句子检索。同时，直接给定相关文档，去除句子检索之前的文档检索环节。所以，最终决定句子检索性能的只有浅层语言分析与句子检索模型两个因素。

在句子检索模型方面采取的是简单的向量空间模型，即采取简单的 TF-IDF 来度量特征词语的权重，如果句子的查询相关度超过一定的阈值 θ，则判定为相关句子并作为检索结果返回。为了进一步简化，θ 设为 0。应该说，我们采取的句子检索模型是最基准的检索方法，并没有引进更复杂的处理技术。通过这样的限制，浅层语言分析成为了句子检索唯一的决定性因素。我们在 TREC2002 与 TREC2003 的数据集上分别作了两组实验，同时引入当年最好的比赛结果及其方法作为对比，从而验证浅层语言分析对句子检索的贡献度。结果如表 9.3 所示。

表 9.3 TREC2002 数据集上的浅层语言分析贡献度评测实验

RunID	句子检索模型	相关技术	平均准确率 P	平均召回率 R	平均 F 值
NOOVEL	简单向量空间模型	浅层语言分析	0.19	0.68	0.257
Thunv1	OKAPI	查询扩展、文档扩展、主题分类、动态阈值调整	0.23	0.34	0.235
NOOVEL2	OKAPI	浅层语言分析	0.19	0.68	0.257

在表 9.3 中，Thunv1(Zhang et al.，2002)是 TREC2002 年新信息检测比赛发布的排名第一的结果。我们发现，文献(Zhang et al.，2002)采用的是 OKAPI 方法并采取了复杂而又比较有效的改进措施，包括查询扩展、文档检索、主题分类以及动态阈值调整等。这代表了当时最好的一个研究水平。为了测试公平性，其他条件不变，同样采用 OKAPI 进行了第二个实验 NOOVEL2，发现结果与 NOOVEL 保持一致。通过测试数据的分析不难发现，采用浅层语言分析技术的优势所在：它的最终性能比历史最高成绩提高了 9.4%。采取同样的条件，我们在 TREC2003 的数据集上进行了第二组实验。结果如表 9.4 所示。

表 9.4 TREC2003 数据集上的浅层语言分析贡献度评测实验

RunID	句子检索模型	相关技术	平均准确率 P	平均召回率 R	平均 F 值
NOOVEL	简单向量空间模型	浅层语言分析	0.59	0.79	0.614
THUIRnv0312	PFM 模型与 bm 2500 相结合	WordNet 的概念扩展结合局部上下文扩展	0.62	0.67	0.564

在上表中，THUIRnv0312(Zhang et al.，2003)是 TREC 在 2003 年发布的最好结果。无论是采用简单向量空间模型还是 OKAPI 检索方法，Noovel 系统的平均 F 值均可以达到 0.614，超过最好水平 8.9%。

通过这两组实验可以得出一个结论：在句子级的检索乃至新信息检测，前期的语言分析预处理比检索算法本身对最终性能的影响还要显著，而人们往往忽视前期的语言预处理。在不准确的语言处理结果基础上，无论在检索建模上怎样努力，都很难取得理想效果，语言处理的中间结果决定了最终性能的上限，更精确的语言处理结果结合一个相对简单的检索方法往往就可以达到甚至超过前者最好的性能。从两组实验来看，在 Noovel 系统中，浅层语言分析至少提高了 9%的最终性能。

另外，从准确率与召回率的指标来分析还可以发现，语言有针对性的准确分析主要贡献在于它更大程度地提高了句子检索的召回率。由于句子自身的特点，已有系统的最大困难就在于难以达到更高的召回率。例如，在 TREC2002 的数据集上，所有系统的召回率均不超过 0.4，而在 TREC2003 数据集上，召回率依然局限在 0.7 以内。浅层句法分析大幅度突破了这些上限。同时，我们还可以看到不同检索方法的准确率差别并不大。

9.4.2　三种句子检索模型的基准实验

下面针对各种检索模型进行实验，同时调节参数与阈值，但不引入查询扩展技术，从而测试出各种方法的基准性能。

1. 向量空间模型

在向量空间模型中尝试不同的权重规格化方法并调节查询相关度的阈值 θ。在 TREC 2003 的数据集上，测试结果如表 9.5 所示。

表 9.5　向量空间模型的句子检索实验(TREC2003 数据集)

规格化方法	阈值	平均准确率 P	平均召回率 R	平均 F 值
余弦/长度	0.0	0.59	0.79	0.614
余弦	0.04	0.59	0.78	0.611
余弦	0.1	0.60	0.51	0.482
长度	0.04	0.57	0.29	0.312
长度	0.1	0.37	0.01	0.023

从表 9.5 可以发现典型的余弦规格化方法比较适合句子检索。从相关度阈值来看，随着阈值提高，准确率提升相当有限，而召回率却急剧下降，导致最终性能受到影响。这说明经过浅层语言分析，某个句子结果中只要包含了查询的特征词语

(哪怕是相关度很小),它很可能就是检索结果。在 TREC2002 的数据集上同样作了几组实验如表 9.6 所示,结论与上面一致。

表 9.6 向量空间模型的句子检索实验(TREC2002 数据集)

规格化方法	阈值	平均准确率 P	平均召回率 R	平均 F 值
余弦	0.0	0.19	0.68	0.257
余弦	0.04	0.18	0.43	0.225
余弦	0.1	0.18	0.40	0.216

另外,对比 TREC2002 与 TREC2003 的测试结果,还可以发现句子级新信息发现的最终性能非常依赖于具体的主题和文档。采用同一种方法,在不同的数据上,最终的性能指标几乎相差三倍之多。

2. 概率检索模型

在 TREC2003 的数据集合上采用 OKAPI 的方法,采取不同阈值,做了三个实验,结果如表 9.7 所示。

表 9.7 概率检索模型的句子检索实验(TREC2003 数据集)

阈值	平均准确率 P	平均召回率 R	平均 F 值
0.0	0.59	0.79	0.614
2.0	0.59	0.79	0.613
8.0	0.61	0.60	0.544

OKAPI 方法计算出来的相关度没有归一化,在句子检索方面的性能与向量空间模型相当。不过已有的文献表明,OKAPI 方法在文档检索方面比简单的向量空间模型更具有优势。

3. 语言模型

采用语言模型计算所有句子生成查询的一元语言概率,然后依据这个概率对句子进行排序,依次取排名靠前的句子作为检索结果。在 TREC2003 的数据集上经过各种对比实验,发现取前 70%的句子集合作为检索结果,能够取得最优的性能,各项指标如表 9.8 所示。

表 9.8 语言模型的句子检索实验(TREC2003 数据集)

阈值	平均准确率 P	平均召回率 R	平均 F 值
70%	0.48	0.63	0.498

文档级的语言模型检索能够达到与向量空间模型相当的性能,但在句子检索

方面，不管怎样对语言模型进行改进，F 值都很难超过 0.5，与向量空间模型概率和模型相比，相差甚远。从理论上不难发现，利用一个句子来构建语言模型存在着本质的缺陷，经过浅层语言分析之后，单个句子包含的词数一般都在 10 个以内，这么少的样本空间来估计语言的概率，数据稀疏问题相当严重，无论采取什么样的数据平滑技术，都很难弥补少量数据所带来的不足。因此可以断定，语言模型不适合句子级的检索，实验也验证了语言模型在句子检索方面的不足。

9.4.3　查询扩展实验

本节给出对比实验测试查询扩展技术在句子检索方面的作用。在 TREC2003 数据上采用向量空间模型，分别完成了如下实验。

(1) NONE：采用简单的向量空间模型，不引入任何其他的查询扩展技术。

(2) WordNet：在 NONE 基础上引入 WordNet 进行语义衍生。

(3) PSEUDO5：在 NONE 基础上引入伪相关反馈，其中相关度排名排名前 5%的句子集合作为伪反馈结果。

(4) PSEUDO15：在 NONE 基础上引入伪相关反馈，其中相关度排名排名前 15%的句子集合作为伪反馈结果。

(5) PSEUDO20：在 NONE 基础上引入伪相关反馈，其中相关度排名排名前 20%的句子集合作为伪反馈结果。

(6) LCE：在 NONE 基础上引入局部共现扩展。

测试结果如表 9.9 所示。

表 9.9　查询扩展技术对比实验(TREC2003 数据集)

实验代码	平均准确率 P	平均召回率 R	平均 F 值
NONE	0.59	0.79	0.614
WordNet	0.59	0.79	0.614
PSEUDO5	0.56	0.88	0.624
PSEUDO15	0.55	0.92	0.631
PSEUDO20	0.55	0.93	0.633
LCE	0.57	0.98	0.643

从上面的测试结果中可以发现。

(1) WordNet 的语义衍生对句子检索没有明显提高，主要原因在于进行语义衍生时进行了限制，因此并没有衍生出很关键的特征词语。另外可能的原因是 WordNet 语义衍生出来的词语更多已属于查询的一部分，它增强了高相关句子的相关度，但对那些相关但相关度很低的句子帮助甚微，可以形象地形容它是一种“锦上添花”而不是“雪中送炭”式的“近亲繁殖”。

(2) 伪相关反馈是一种较好的查询扩展方法，在不特别损失检索准确率的前提下，它极大地提高了检索的召回率，从而提高了整体性能。另外，随着反馈句子数目的增加，性能也在逐渐地提高，但增长幅度在缩小。同时，我们的研究还发现超过 20%的反馈比例就很难再提高系统性能，继续增加反馈比例只会起到副作用。当然，具体的百分比例和具体的数据集有相当大的关系，不能一概而论。

(3) 局部共现扩展尽管缺乏很严格的理论依据，但是可以通过统计的方法挖掘出特征词语之间的内在关联性，并能够较好地改进查询的结果。在我们的实验中取得了好的性能，这是一种有潜力的查询扩展思路。

按照综合指标平均 F 值，将各种方法进行综合排序，结果如图 9.5 所示。

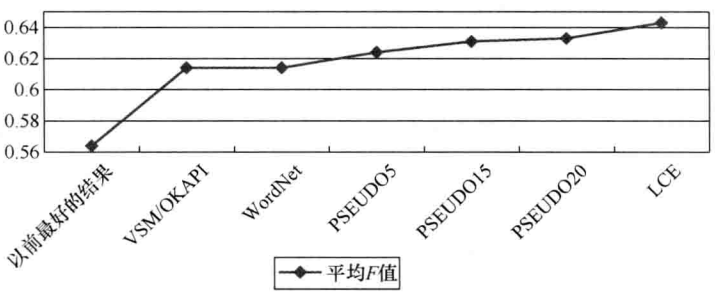


图 9.5 各种句子检索技术的性能对比

9.5 新信息检测

含有所谓“新信息”的句子一般具有两个特点，即内容相关性和时序新颖性。内容相关性指句子的内容必须与主题密切相关。时序新颖性指它传达了以前句子所没有涵盖的新信息。假定用户在检索之前，对当前主题的信息一无所知，所有的信息都来源于检索结果，因此时序意义上的第一个相关句子默认为带有“新信息”的句子。新信息检测形式化定义为 ND (Q, $<s'_1, s'_2, \cdots, s'_{x-1}>, s'_x$) = TRUE，当且仅当 s'_x 是包含新信息的句子，否则 ND(Q, $<s'_1, s'_2, \cdots, s'_{x-1}>, s'_x$) = FALSE，其中$<s'_1, s'_2, \cdots, s'_{x-1}>$为 s'_x 以前出现的新信息句子或相关句子，ND 为对应具体的新信息检测方法(这里讨论的句子都是经过浅层语言分析之后去除噪声的结果且均与主题相关)。

本节引入新颖度的概念来量化一个句子在特定时间序列上的新颖程度，从而可根据新颖度来判别句子是否含有新信息。在已有研究成果的基础上，提出了三种新颖度的评价方法，即词重叠度及其扩展方法、相似度比较方法、信息增强的评价方法。随后，阐述了一些计算方法，如语言模型与语义距离计算的方法，最后在 TREC2003 的数据集上进行新信息检测实验，并给出了分析结论。

9.5.1　词重叠度及其扩展

1. 基于词重叠度的句子新颖度计算

词重叠度是一种简单而有效的句子新颖度衡量方法。给定某个句子 S_i，则其前面句子 S_j 与 S_i 的词重叠度可以定义为

$$\text{Overlap}(S_i \leftarrow S_j) = \frac{|S_i \cap S_j|}{|S_i|}$$

这里将句子看成是词的集合，词重叠度表示的是两个句子共同使用的特征词数在当前句子中所占的比重。Overlap 的计算具有时序方向性，即 $\text{Overlap}(S_i \leftarrow S_j) \neq \text{Overlap}(S_j \leftarrow S_i)$。Overlap 的取值范围为[0.0, 1.0]，取值越大，则句子 S_j 与 S_i 越接近，拥有的共同信息越多(即新信息越少)。若 Overlap 为 0，两个句子没有任何相同的词语，则句子 S_i 包含的都是新信息；若 Overlap 为 1，则表示两个句子完全相同，不存在新的信息内容。基于词的重叠度，这里提出了一种句子新颖度的衡量方法，即

$$\text{OverlapNov}(S_i) = 1 - \max_{0<j<i}\{\text{Overlap}(S_i \leftarrow S_j)\}$$

首先计算当前句子与以前所有相关句子的词重叠度，选择重叠度最大的结果作为当前句子与历史结果的信息重叠度，最后通过减法，计算出句子的信息新颖度。同样，基于词语重叠度，还可以给出另外一种计算形式，即

$$\text{OverlapNov2}(S_i) = 1 - \text{Overlap}(S_i \leftarrow \bigcup_{0<j<i} S_j)$$

这种方法将以前所有的相关句子视为一个整体以便于计算。但是信息并不是词语的简单叠加，这样做可能缺乏必要的理论依据。以下面三个时序上连续的句子为例。

句子 S_1：Tom likes Jerry.

句子 S_2：Jerry likes Jack.

句子 S_3：Jack likes Tom.

按照 OverlapNov2 的定义，计算过程如下。

$$\begin{aligned}
&\text{OverlapNov2}(S_3)\\
&=1-\text{Overlap}(S_3 \leftarrow \bigcup_{0<j<3} S_j)\\
&=1-\frac{|\{\text{Jack},\text{like},\text{Tome}\} \cap \{\{\text{Tome},\text{like},\text{Jerry}\} \cup \{\text{Jerry},\text{like},\text{Jack}\}\}|}{|\{\text{Jack},\text{like},\text{Tome}\}|}\\
&=1-\frac{|\{\text{Jack},\text{like},\text{Tome}\} \cap \{\{\text{Tome},\text{like},\text{Jerry},\text{Jack}\}\}|}{|\{\text{Jack},\text{like},\text{Tome}\}|}\\
&=1-\frac{|\{\text{Jack},\text{like},\text{Tome}\} \cap \{\{\text{Tome},\text{like},\text{Jerry},\text{Jack}\}\}|}{|\{\text{Jack},\text{like},\text{Tome}\}|}
\end{aligned}$$

$$=1-\frac{|\{\text{Jack},\text{like},\text{Tome}\}|}{|\{\text{Jack},\text{like},\text{Tome}\}|}$$
$$=0$$

计算结果是信息新颖度为 0，而实际上句子 S_3 传达了 S_1 和 S_2 所没有的新信息内容，计算方法 OverlapNov2 的缺陷在于它将信息内容进行简单叠加，另外没有考虑词语位置的先后顺序。虽然计算方法 OverlapNov 也忽略了位置信息，但它将一个单句作为研究对象，避免了上面例子中存在的问题。

2. 带权重的词重叠度计算

词重叠度方法简单、直接而有效，很多研究者采用词重叠度达到了很好的效果。但是在 Overlap 的计算中，它将句子中所有的词语都平等对待，没有进行区分，实际上这是不科学的。以 TREC 数据集主题 N51 的句子，Pinochet cast a long and wide shadow in economic affairs as well , launching a privatized social security system and other free market policies that set examples that are still models from Argentina to Mexico 为例，其中的“long”、“wide”等词语对新信息的区分度远小于“economic”、“security”、“policies”等词。为此引入了词的 χ^2 统计值 chi_w 作为词权重并对词重叠度进行扩展，即

$$\text{Overlap}^*(S_i \leftarrow S_j)=\frac{\sum_{w\in S_i\cap S_j}\text{chi}_w}{\sum_{w\in S_i}\text{chi}_w}$$

可以看出，当词语 χ^2 统计值 chi_w 相等的时候，计算方法 Overlap* 就退化为 Overlap。

9.5.2 相似度比较方法

词重叠度方法考虑了当前句子与历史信息的词重叠度，而没有兼顾当前句子与主题的相关性。为此，需要对当前句子与主题和历史信息的相关性进行比较，其目的在于选择与主题相关度更高而与以前句子相关性更低的句子作为承载新信息的结果。依据历史信息处理的不同策略，这里提出三种比较方法来衡量句子的新颖程度，即

$$\text{MaxPrevNov}(S_i)=\lambda\text{Sim}(S_i,T)-(1-\lambda)\max_{k<i\wedge S_k\in R}\text{Sim}(S_i,S_k)$$
$$\text{AverPrevNov}(S_i)=\lambda\text{Sim}(S_i,T)-(1-\lambda)\text{AVG}_{k<i\wedge S_k\in R}\text{Sim}(S_i,S_k)$$
$$\text{PrevAverNov}(S_i)=\lambda\text{Sim}(S_i,T)-(1-\lambda)\text{Sim}(S_i,\overline{\text{SP}})$$

其中，λ 为主题相关度与历史相关度的调节参数($0\leqslant\lambda\leqslant1$)；$R$ 为相关句子集合；T 为表示主题的查询向量；$\overline{\text{SP}}$为 S_i 以前所有句子的平均向量；MaxPrevNov 表示当前句

子 S_i 与历史相关度最大值的差值；AverPrevNov 表示当前句子 S_i 与历史相关度平均值的差值；PrevAverNov 表示当前句子 S_i 与历史平均句子向量相关度的差值。

目前，研究者常使用的最大区间相关度就属于这类方法，和 MaxPrevNov 的思路一致。

9.5.3　信息增强评价方法

词重叠度与相似度都是直接对一个单句进行计算，但单句长度有限，实际计算时往往容易发生大的偏差，而在句子集合或者文档上计算则情况要好得多。为此，这里提出了一种信息增强的评价方法，比较引入当前句子前后信息的增强情况，从而间接估算出该句子的新颖程度。假如当前句子没有新的信息，则增加该句子并不会对整体信息带来收益；反之，则丰富了原来的语义。这比直接计算单个句子的信息要宽泛得多，理论上更有说服力。基于相关度，可以得到信息增强评价方法为

$$\mathrm{IGNov}(S_i)=\mathrm{Rel}(T,\mathrm{PrevSentences}+S_i)-\mathrm{Rel}(T,\mathrm{PrevSentences})$$

其中，PrevSentences 是 S_i 以前的所有相关句子的并集；T 为表示主题的查询向量；Rel 为主题与句子集合的相关度，可以采用第 4 章提到的检索模型进行计算，同样可以采用语言模型条件概率的方式进行估计，即

$$\mathrm{IGNov}'(S_i)=P(T|\mathrm{PrevSentences}+S_i)-P(T|\mathrm{PrevSentences})$$

最后，我们还可以采取类信息熵的办法来估计，即

$$\begin{aligned}\mathrm{IGNov}''(S_i)=&P(T|\mathrm{PrevSentences})\log P(T|\mathrm{PrevSentences})\\&-P(T|\mathrm{PrevSentences}+S_i)\log P(T|\mathrm{PrevSentences}+S_i)\end{aligned}$$

需要说明的是，这里提到的信息增强和机器学习方面的信息增益不是同一个概念，采用的计算方法也不一样，但出发点是一致的。

9.5.4　其他方法

1. 语言模型

可以采用 Kullback-Leibler 等(1951)交叉熵来度量两个语言模型 Θ_1 与 Θ_2 之间的距离，即 $\mathrm{KL}(\Theta_1\,||\,\Theta_2)=\sum_w p(w|\Theta_1)\log\dfrac{p(w|\Theta_1)}{p(w|\Theta_2)}$。通过 KL 距离，可以定义句子 s_i 与以前所有相关句子的新颖度，即

$$\begin{aligned}\mathrm{KLNov}(s_i|s_1,\cdots,s_{i-1})&=\mathrm{KL}(\Theta_{s_i}\,||\,\Theta_{s_1\cdots s_{i-1}})\\&=\sum_w p(w|\Theta_{s_i})\log\frac{p(w|\Theta_{s_i})}{p(w|\Theta_{s_1\cdots s_{i-1}})}\end{aligned}$$

语言模型估计时存在数据稀疏问题，采用最大似然估计与插值平滑技术，可以计算其中的参数(Leah et al.，2002)，即

$$p(w \mid \Theta_{s_i}) = \lambda_1 p(w \mid \Theta_{\mathrm{ML}s_i}) + (1-\lambda_1) p(w \mid \Theta_{\mathrm{ML}s_1,\cdots,s_i})$$

$$p(w \mid \Theta_{s_1,\cdots,s_{i-1}}) = \lambda_2 p(w \mid \Theta_{\mathrm{ML}_{s_1,\cdots,s_{i-1}}}) + (1-\lambda_2) p(w \mid \Theta_{\mathrm{ML}s_1,\cdots,s_i})$$

也可以采取退化的平滑技术来估计参数[James, 2003],即

$$p(w \mid \Theta_{s_i}) = \lambda_s p(w \mid \Theta_{\mathrm{ML}s_i}) + \lambda_t p(w \mid \Theta_{\mathrm{ML}t}) + \lambda_e p(w \mid \Theta_{\mathrm{ML}e})$$

其中,$\Theta_{\mathrm{ML}s_i}$为句子 s_i 的最大似然语言模型;$\Theta_{\mathrm{ML}t}$为句子 s_i 所在的主题范围内的最大似然语言模型;$\Theta_{\mathrm{ML}e}$为英语通用语言环境下的最大似然语言模型。

经过实验,语言模型的方法能够取得的性能与相似度比较方法相当。

2. 句子语义距离计算

句子语义距离计算旨在量化语义或者概念表达式之间的不同程度,从而间接地计算出句子语义相似度。采用 WordNet 作为语义知识库,提出了一种语义距离计算的方法来度量句子信息的新颖程度。仅考虑语义的上下文关系,WordNet 是一种树型结构,语义的孩子节点(也就是一个概念 Synset)c 与其父亲节点 p 之间的语义距离可简单地估计为

$$\mathrm{Dist}(c,p) = \mathrm{IC}(c) - \mathrm{IC}(p)$$

其中,IC(c)表示的是概念 c 的信息量,即

$$\mathrm{IC}(c) = -\log P(c) = -\log\{\sum \mathrm{freq}(w)/N\}$$

其中,w 属于集合 c 或者 c 所有的下位 Synset;freq(w)为 w 的频率;N 为总频率,这些都可以从 WordNet 提供的平衡语义语料库中训练得到。

基于此可以计算出任意两个语义节点 c_1 与 c_2 之间的语义距离,即

$$\mathrm{Dist}(c_1,c_2) = \mathrm{IC}(c_1) + \mathrm{IC}(c_2) - 2\times \mathrm{IC}(\mathrm{LSuper}(c_1,c_2))$$

其中,LSuper(c_1,c_2)表示的是两个语义节点 c_1 与 c_2 最近的共同上位概念,其意义类似于最小公倍数。

如图 9.6 所示,LSupper("brother", "arm")="Person",而 Dist("brother", "arm")实质上就是图 9.6 中所有虚线权重之和。根据 IC(c)定义,可以推算出 $\mathrm{Dist}(c_1,c_2) \geqslant 0$。

某个词语往往存在多个语义 Synset,在计算词语之间的距离时,并不进行词义排歧,即

$$\mathrm{Dist}(w_1,w_2) = \max_{w_1\in c_1 \wedge w_2 \in c_2}\{\mathrm{Dist}(c_1,c_2)\}$$

仅考虑词语之间的上下位关系是不够的,我们还进一步考虑了同义、部分整体等关系。在词语距离的基础上,借助于几何学上关于点到直线距离的原理,提出了词 w 与句子 S 距离 WSSD(w, S)的计算方法,即

$$\mathrm{WSSD}(w, S) = \min\{\mathrm{Dist}(w,w_i) \mid w_i \in S\}$$

其中 w 与 w_i 是词;S 为句子。

在此基础上定义一个句子 S_1 与另一个句子 S_2 之间的语义距离 SSSD,即

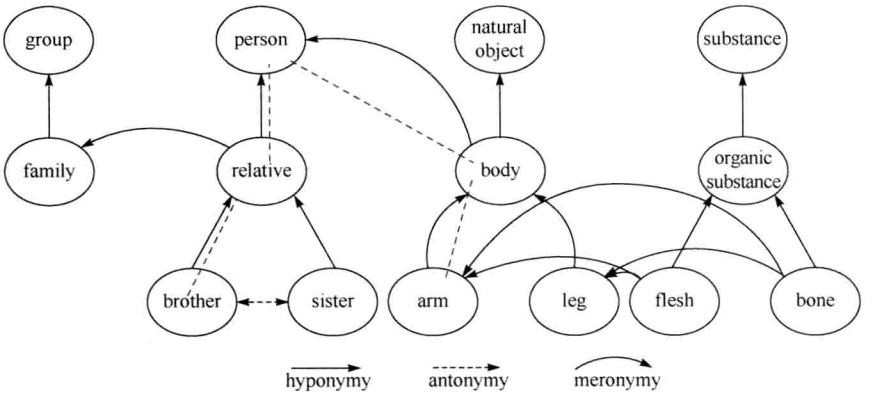


图 9.6　WordNet 中语义距离计算示意图

$$\mathrm{SSSD}(S_1,S_2)=\frac{\sum_{w_i\in S_1}\mathrm{WSSD}(w_i,S_2)+\sum_{w_j\in S_2}\mathrm{WSSD}(w_j,S_1)}{|S_1|+|S_2|}$$

其中，$|S_1|$与$|S_2|$为句子中包含的词数，根据表达式可以知道 SSSD 满足交换律，即 $\mathrm{SSSD}(S_1,S_2)=\mathrm{SSSD}(S_2,S_1)$。

图 9.7 给出了句子语义距离计算的原理图，可以定义基于语义距离计算的句子新颖度，即

$$\mathrm{SDNov}(S_i,S_{1,\cdots,i-1})=\lambda_1\mathrm{SSSD}(S_i,T)-\lambda_2\mathrm{SSSD}(S_i,S_{1,\cdots,i-1})$$

其中，$\mathrm{SSSD}(S_i,T)$表示的是句子 S_i 与主题 T 的语义距离；$\mathrm{SSSD}(S_i,S_{1,\cdots,i-1})$表示的是句子 S_i 与历史信息 $S_{1,2,\cdots,i-1}$的语义距离；λ_1 与 λ_2 为调节参数。

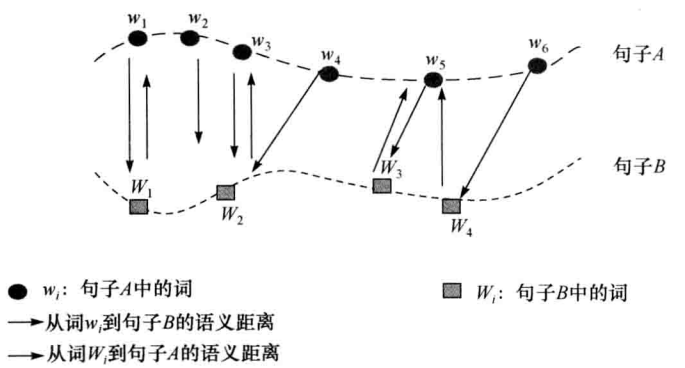


图 9.7　句子语义距离计算的原理图

9.5.5　新信息检测实验与分析

在 TREC2003 的 50 个主题数据上，采用不同的新信息检测技术，经过多次参

数调节，进行了多组实验，表 9.10 给出了 11 组有代表性的实验结果。

表 9.10 新信息检测实验(TREC2003 数据集)

新信息检测方法	调节参数	平均准确率 P	平均召回率 R	平均 F 值
MaxPrevNov	0.7	0.50	0.16	0.185
AverPrevNov	0.7	0.39	0.75	0.462
PrevAverNov	0.7	0.23	0.02	0.031
MaxPrevNov	0.9	0.40	0.71	0.462
AverPrevNov	0.9	0.39	0.79	0.470
PrevAverNov	0.9	0.40	0.59	0.419
OverlapNov	0.0	0.47	0.55	0.443
OverlapNov2	0.0	0.39	0.79	0.470
IGNov	0.5	0.37	0.66	0.426
KLNov	0.0	0.39	0.61	0. 410
SDNov	0.9	0.38	0.58	0.389

从表中可总结出如下结论。

(1) 前面 6 组是相似度比较方法，调节参数 λ 由 0.7 升为 0.9，各种相似度比较方法都提高了性能，尤其是 PrevAverNov 的平均 F 值由 0.031 跃升为 0.419。这意味着一个句子是否传达新的信息，一方面和历史信息的比较有关，其本身与主题的相关度更为关键。一个句子和主题的相关度很高的时候，它往往就是一个新信息的载体。

(2) 相比较而言，在相似度比较方法中 AverPrevNov 较优。

(3) OverlapNov 和 OverlapNov2 是简单直接的词重叠度计算方法，但是都取得了较优的结果，这表明词重叠方法简单而有效。

(4) 对比 OverlapNov 和 OverlapNov2，OverlapNov2 提高了 6.1%，这说明带权重的词重叠度更能客观地反映新信息的实际情况。

(5) 信息增强评价方法 IGNov 取得了一定的效果，性能比较偏低，这说明尽管计算更加便利，但是引入了全部历史信息，增大了误差，导致了最终结果没有更好地反映信息的新颖程度。

(6) KLNov 和 SDNov 作为一个实验性质的方法，有一定的理论依据，也可以取得一定的效果，但是仍然无法实用。KLNov 仍然受到数据稀疏的困扰，而 SDNov 过分依赖于语义体系，而且计算代价非常高。在我们的实验中，简化之后的句子语义距离的计算复杂度依然很高。

9.6　监督学习条件下的句子检索与新信息检测

前面介绍的是非监督学习条件下的句子检索与新信息检测，没有任何先验知识或者结果可用来学习和训练。非监督学习环境是新信息检测中绝大多数的应用场景，也是技术实用化要面对的挑战。在某些情况下，用户可能会给出部分的提示或反馈，因此可以事先获得部分结果，并据此采取机器学习的方法进行调整，以期进一步提高新信息检测的性能。

9.6.1　监督学习环境下的参数调整与阈值设置

在监督学习条件下，可以利用已有的知识或部分结果，通过机器学习的方法，进一步提高新信息检测的性能。

1. 进一步的特征选择

可以将已知的相关句子作为一个结果集合，而其他的不相关句子作为对立的结果集合，利用互信息或 χ^2 统计方法做进一步的特征选择，排除噪声数据，保留更有区分能力的特征词语。这为后续的句子新信息检测构造了更优的特征空间。

2. 真实反馈

在监督条件下，给定的结果是真实的反馈。图 9.8 给出了相关部分结果，对应部分剩余的句子就是不相关的句子集合（可分别视为正向反馈和逆向反馈），采取经典的 Rocchio 公式对原始查询进行扩展，即

$$Q^r = \alpha \times Q + \beta \times \frac{\sum S_{\text{rel}}}{N_{\text{rel}}} - \gamma \times \frac{\sum S_{\text{nrel}}}{N_{\text{nrel}}}$$

其中，N_{rel} 为已知的相关句子数；N_{nrel} 为已知的不相关句子数；α、β 和 γ 分别为各部分的权值（属可调参数）。

最终的查询实现了部分结果的反馈，更好地适应了应用环境。

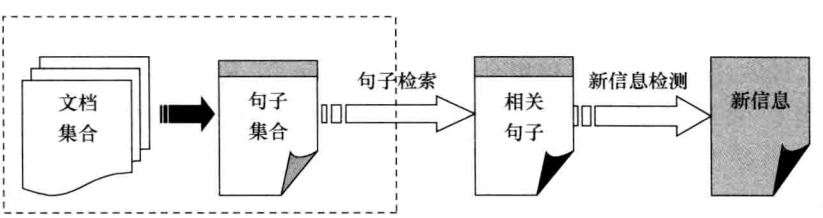


图 9.8　监督学习条件下的新信息检测（给定部分文档中的相关句子与新信息句子）

3. 参数调整

句子检索与新信息检测的算法模型往往都存在部分可以调节的参数。在非监督条件下，参数更多的是在某个数据集合上通过实验基础上得到的经验值，而新信息检测往往依赖于具体的主题与文档集，因此原来的参数并不一定适用于新的数据集。在监督条件下，给定具体主题的部分结果，可以据此有目的地调节参数，使之尽可能地符合当前主题的实际情况，从而更好地指导剩余结果的分析。

4. 阈值设置

阈值的设置直接决定最终的结果集合。和参数调整类似，阈值设置存在同样的问题。在监督条件下，阈值存在一般性的规律。以图 9.9 所示的情况为例，此时给定了所有的相关句子，并已知每个主题前 5 个文档具有新信息的句子。在某个范围的句子集合内，定义一个参量 Ratio_N_R 为

$$\text{Ratio_N_R}=\frac{\text{新信息句子数}}{\text{相关句子数}}$$

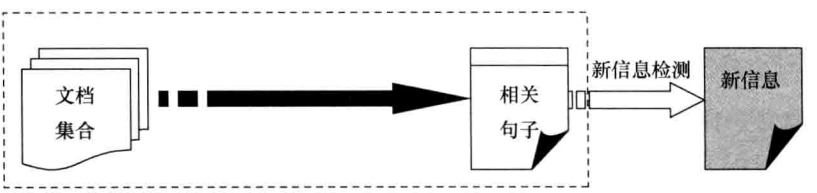


图 9.9 监督学习条件下的新信息检测(给定所有相关的句子与部分文档中新信息的句子)

对于每个主题，在前 5 个文档范围内逐步扩大句子集合，每增加 1 个带新信息的句子，记录此时的 Ratio_N_R，对于每个不同的主题来说，Ratio_N_R 千差万别，不存在一般性的规律。但是对于同一个主题来说，随着文档叙述文字的逐渐深入与展开，相关句子中包含新信息的几率愈来愈小，Ratio_N_R 存在逐渐变小的趋势。通过曲线拟合的方法，可以通过早期的 Ratio_N_R 来估计内在的变化趋势，最后计算出整个文档集合范围内的 Ratio_N_R，给定所有的相关句子数目，就可以大致估算出最后新信息句子的总体数目。图 9.10 给出了在 TREC2003 数据集合上 Ratio_N_R 的拟合曲线，其中 x 轴表示的是前 5 个文档范围内的 Ratio_N_R，y 轴表示的是整个文档范围内的 Ratio_N_R，分别采用了一次函数、二次函数、对数函数、指数函数、幂函数进行模拟。最后发现，对数函数 $y=0.1168\ln x+0.2995$ 的平方误差最小，50 组数据平方误差的总和

为$R^2=0.4567$。

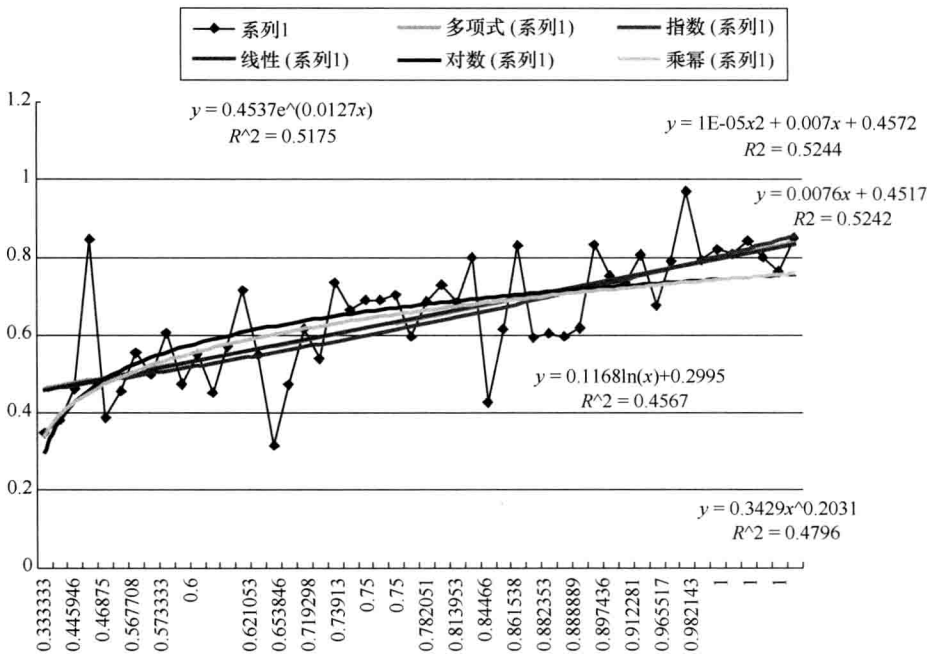


图 9.10　TREC2003 数据集上 Ratio_N_R 的拟合曲线

通过这种方法能大致估计出输出的结果数目，因此可以按照新颖度进行排序，从高到低输出所需数量的句子结果，从而避免了非监督条件下阈值设置的盲目性。

9.6.2　基于分类的句子检索与新信息检测方法

在监督条件下可以采取分类的思想实现句子的检索或新信息检测，基本思想是在给定的部分文档结果中，将相关句子视为类别 REL，将不相关的句子视为 NON-REL，训练出一个二元分类器，对剩余文档中的句子依次分类，将它们归结为相关的 REL 类或者不相关的 NON-REL。同样，在新信息检测环节，可以将新信息的句子视为 NEW 类，而相关而不包含新内容的句子归为 NON-NEW 类，具体的训练过程和分类过程一致。

以 TREC2003 子任务三的句子检索为例(这里举例的数据都来自第一个话题 N1)。给定了前 5 篇的相关句子信息，因此可以将其中的 52 个相关句子作为 REL 类，而其他的 106 个句子为对应的 NON-REL 类，为此可以得到 158 个训练样本。表 9.11 给出了部分训练样本与测试数据片段。

表 9.11 相关性分类的训练样本与测试数据片段

句子 ID	向量表示	类别
NYT19980629.0465 8	482：0.329448 30418：0.768812 85128:0.548081	REL
NYT19980629.0465 9	482:1.000000	NON-REL
NYT19980629.0465 10	482：0.226595 5501：0.287315 14602:0.513173 51265:0.431760 66307:0.436277 76766:0.367625 82458:0.301422	NON-REL
NYT19980629.0465 11	482：0.489471 485：0.457421 5501：0.310316 44565：0.421367 55411:0.405021 79280:0.336597	REL
APW19990924.0219 6	3446：0.342615 3848：0.425703 12612:0.401669 15879:0.250162 36578:0.434427 47512:0.309547 82290:0.439235	UNKNOWN(未知,待分类)
APW19990924.0219 7	56001:0.445543 82428:0.771948 82458:0.453418	UNKNOWN(未知,待分类)

在表 9.11 中,每维句子向量的格式表示为“特征词语 ID:权重”,前四行属于训练样本,后两条是待分类的测试数据。如何根据已有的训练结果,来推断测试数据的归属,这是一个典型的分类问题,这里给出的原型系统 Noovel 采用了支持向量机 SVM 二元分类方法。具体实现上,采用了一套开放源代码系统 SVM^{light} (Version 5.00)。针对句子检索与新信息检测,都训练了两种 SVM 分类器。其中一种直接采用已知的部分结果训练分类器(称之为 Inductive SVM),另外一种将测试数据作为未知分类结果也加入到训练样本中,一起参与机器学习训练(称之为 Transductive SVM)。后者对小样本分类很有帮助,比较适合测试数据分布特性和训练样本相似的分类问题。

9.6.3 实验与分析

依据不同的监督条件,在 TREC2003 的数据集上做了两组监督条件下的实验。

1. 监督实验 1

给定所有的相关句子,可以直接在相关句子上检测出带有新信息的句子,实验

结果如表 9.12 所示。

表 9.12　给定所有相关句子的新信息检测实验(TREC2003 数据集)

新信息检测方法	平均准确率 P	平均召回率 R	平均 F 值
MaxPrevNov	0.67	0.73	0.684
AverPrevNov	0.65	0.79	0.697
OverlapNov	0.71	0.74	0.704
OverlapNov2	0.65	1.00	0.774
IGNov	0.65	0.99	0.770
IGNov+反馈	0.65	1.00	0.774

对比非监督条件下的新信息检测(表 9.10),给定相关句子的条件下,新信息检测的性能至少提高了 45.5%(表 9.10 中平均 F 值最好的结果为 0.470,而这里最差的性能为 0.684),由此可以估计出句子检索所带来的损失。

将非监督条件下的新信息检测 F 值除以任务 2 对应的指标,大致可以推算出句子检索的性能指标。如同样采取 OverlapNov2 方法,不同条件下的性能比为 0.470∶0.774=0.607,这正好是句子检索 F 值大致的性能范围。

最后两组数据能够表明反馈发挥的作用,方法之间的对比与非监督条件类似,这里不再赘述。

2. 监督实验 2

给定前 5 篇文档中所有的相关句子,在剩下的 25 篇文档中,测试此时的句子检索性能。句子检索的实验结果如表 9.13 所示。

表 9.13　给定前 5 篇文档相关句子的句子检索实验(TREC2003 数据集)

句子检索方法	平均准确率 P	平均召回率 R	平均 F 值
VSM	0.61	0.82	0.664
OKAPI	0.60	0.81	0.657
支持向量机分类	0.59	0.71	0.627
VSM+反馈	0.58	0.96	0.701

对比非监督条件下的句子检索,给定部分结果的前提下,句子检索的性能均得到了提高,参数调整与阈值设置起到了一定的作用。

支持向量机分类的方法能够部分地解决句子检索的问题,但并不是最优的方案。一方面,支持向量机的训练过程非常的耗费时间和空间,Transductive 方式的支持向量机往往需要 5～6 个小时才能完成小样本的训练。另一方面,支持向量机的最终性能并不如普通的向量空间模型。主要原因在于分类器主要是依据训练样本分布来计算一个最优的分类面,忽略了句子检索过程中很重要的查询需求,而向

量空间模型直接计算句子与查询向量的相关度，更能反映句子检索的内在规律。

9.7 本章小结

本章介绍了新信息检测的需求背景，主要围绕句子级别信息检索与新信息检测的总体构架展开研究。本章主要探讨了如下内容。

(1) 讨论了句子检索的主要建模方法，包括向量空间模型、概率检索模型与语言模型。为了克服句子的局限性，引入了查询扩展的技术，包括借助于 WordNet 的语义衍生扩展、伪相关反馈扩展和采用高频共现词语的局部共现扩展等。实验表明，在浅层语言分析的基础上，简单向量空间模型可达到较好的结果。受语义资源和分析深度的限制，当前阶段的语义扩展作用有限。伪相关反馈与局部共现扩展都能够帮助提高句子检索的性能，局部共现扩展是很有潜力的查询扩展与文档扩展的技术。在相关句子结果的基础上，对信息新颖度进行建模量化，主要提出带权重的词相关度、相似度比较以及信息增强方法，讨论监督条件下的机器学习方法，从而能够进一步提高新信息检测的性能。

(2) 句子级别的新信息检测，提出三种具有代表性的信息新颖度的量化方法，包括词重叠度及其扩展方法、相似度比较方法和信息增强的评价方法。目的在于兼顾信息与主题的相关性，同时还要与已有历史的信息进行比对，寻找新信息之所在。

(3) 在监督条件下如何进行机器自动学习并调整参数，主要手段包括特征选择、真实反馈、调整参数和阈值设置，同时提出基于分类的句子检索与新信息检测方法。

在开放测试数据集合上进行了大量实验，并从理论上对实验结果进行了各种对比分析。最终将研究的理论与技术综合地集成到了 Noovel 系统中，这是一个可以灵活配置，融入了多种技术的实验平台。其中，将各个阶段的分析结果保留到中间结果文件，便于各个研究阶段在较高的相同水准上，进行各种对比实验。

相关工作与扩展阅读

经典的向量空间模型由 Salton 等于 20 世纪 60 年代末提出，并成功地应用于著名的 SMART 文本检索系统，但向量空间模型的方法并不局限于这个模型自身。这实际上是一种通用的文档信息表示方法，任何一种检索模型都可以利用它的概念来表示文档，利用它的方法来辅助检索。向量空间模型将文档视为向量，向量的每一维代表一个信息表达元素。它可以是汉语中单个的字，也可以是任何语言中的一个词或者短语，也可以是没有实际意义的 n 元组。这些元素通过对数据集合中的每一个文档进行处理得到。

在 Noovel 系统中，每一个文档或句子经浅层语言分析处理后得到一系列的特征词语 ID，它们就可以构成一个特征空间，每个词语 ID 就是特征空间中的一维。文档可以用来自文档中的那些词语 ID 来表示，也就是说可以把文档看成是这个空间中的向量，每一维对应一个词语 ID。因此，文档被转化为文档向量空间上的一个点。给定一个用户查询 Q 与文档集合中的文档 D，概率模型通过估计用户查询 Q 与文档 D 相关的概率来判定 D 是否与 Q 相关。概率模型假设这种概率只取决于查询串和文档，这样就将检索问题转化为条件概率 $\mathrm{Prob}(R \mid D,Q)$ 的求解问题，即

If $\mathrm{Prob}(R \mid D,Q) > \mathrm{Prob}(\mathrm{NR} \mid D,Q)$

Then D 属于相关文档集合 R（即检索结果）

Else D 属于不相关文档集合 NR（即不是检索结果）

概率检索模型最早由 Maron 和 Kuhns 于 1960 年提出（Maron，1960），经过发展，概率检索模型已经从概念走向实际应用。一般概率检索模型包括二值独立标引模型、二值独立检索模型、统一概率模型、Bayesian 信息检索模型、二维泊松模型、Bayesian 专家观点组合（张文进，2005）等，经过改进概率检索模型又出现了推理网络模型和信任度网络（欧亮，2011）。模型概率检索模型的主要优点是文献根据它们与检索的相关概率值的大小按照递减顺序进行排序，其缺点是参数估计难度较大（王树梅，2007）。基于概率检索模型的 OKAPI 系统在多届 TREC 中取得了优异成绩（OKAPI 系统是 Robertson 等在伦敦城市大学研究和开发的一个著名的概率检索系统），其理论成果已经被广为接受（Robertson，1994；Robertson 1996）。它基于概率检索模型设计，使用著名的 BM 25 公式及其变形等计算特征词语的权值，这种权重计算公式在概率模型中已经成为公认成熟的标准计算方法。

参 考 文 献

欧亮．2011．信息检索可视化研究综述．情报探索，3：83-87.

王树梅．2007．信息检索相关技术研究．南京理工大学博士学位论文．

张文进．2005．文本信息检索中的概率模型．情报杂志，3：107-110.

Chen S F，Goodman J T. 1996. An empirical study of smoothing techniques for language modeling// Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics

James A，Wade C，Bolivar A. 2003. Retrieval and novelty detection at the sentence level// Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.

Kullback S，Leibler R A. 1951. On information and sufficiency. Annals of Mathematical Statistics，22：79-86.

Leah S，Allen J，Magaret E，et al. 2002. UMass at TREC 2002：cross language and novelty

tracks// Proceedings of the Eleventh Text Retrieval Conference.

Maron M E, Kuhn J L. 1960. On relevance, probabilistic indexing and information retrieval. Journal of the ACM: 216-244.

Robertson S E, Walker S, Beaulieu M M, et al. 1996. Okapi at TREC-4// Proceedings of the 4th Text REtrieval Conference (TREC-4), 236: 73-86.

Robertson S E, Walker S. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval// Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval: 232-241.

Zhai C, Lafferty J. 2001. A study of smoothing methods for language models applied to ad hoc information retrieval// Proceedings of the 24th Annual International ACM SIGIR Conference.

Zhang M, Lin C, Liu Y, et al. 2003. THUIR at TREC 2003: novelty, robust and web// Proceedings of the Twelfth Text Retrieval Conference.

Zhang M, Song R, Lin C, et al. 2002. Expansion-based technologies in finding relevant and new information: THU TREC2002 novelty track experiments// Proceedings of the Eleventh Text Retrieval Conference.

第 10 章　人物追踪中的数据预处理与属性抽取

10.1　概　　述

人物是抽象对象的代表之一，对其进行挖掘代表了大数据文本挖掘技术的研究方向之一。作为大数据搜索与挖掘的实际应用，本章讨论人物追踪中的数据预处理与属性抽取方法。一方面，人物追踪要解决人物相关信息的同一性判断问题、数据融合问题和跟踪问题，将为与人物处于同一抽象层次的其他对象，如事件、地点、机构等的挖掘提供技术上的参考，也将为抽象对象挖掘本身凝练理论和方法。另一方面，人物追踪的研究还可以产生一些特点应用。

10.1.1　研究背景

搜索人物时，歧义现象是十分普遍的。这些歧义可大致分为两类：一类是不同人物及其他非人物实体可能具有相同的名字；另一类是同一个人物可能具有不同的名字或称谓(Taffet, 2004)，这样的歧义可能造成人物搜索的结果不理想。虽然可以在人名后边添加属性，通过组合查询来细化结果，但这样就违背了搜索引擎所追求的简单快捷原则，且用户往往不是精通搜索技巧的专家，他们也往往不愿意提交额外的查询词。一项调查表明，当人们在 AlltheWeb. com 和 Alta Vista 上搜索人名的时候，愿意提交额外查询词的情况分别只占 11. 4%和 17. 5%(Spink et al. , 2004)。在一般搜索引擎上搜索名人时，遇到的同名歧义现象是比较少的。究其原因，可能是因为一般的搜索引擎都会根据网页的重要程度对返回的结果进行排序。在同名人物的网页中，名人的网页对非名人的网页具有“淹没”效果。如果关注的正好是名人，那么返回结果可能符合我们的需求，但如果想了解一个人物都担任过什么职务、现在多大年龄、经常跟哪些人交往最多、参加过哪些活动、各个活动之间有什么样的联系等，返回的网页列表往往很难让人满意。

在大数据文本空间内，对人物的跟踪也应该按照相关的线索来进行，如人物的横向(空间)跟踪(人与人的联系、人与社会的联系等)和人物的纵向(时间)跟踪(人的职位变化、人的处所变化、人的态度立场变化、人的生活状态变化等)。人物追踪就是要把不同文档中出现的人物信息片段有效地组织起来，呈现人物信息的全貌，反映该人物的生活历程。为此，在大数据文本处理领域，人物追踪在内容上可大体分为两大部分，即制作人物的履历表、对人物的事件进行组织。其中，制作人物履历表就是从大数据文本中自动提取出该人物的属性信息，并把分散的各种属性信

息进行整合，形成完整的生平履历；对人物的事件进行组织就是把人物参与的相关事件的报道按照内在发展逻辑有效地组织起来，并形成清晰的逻辑结构。其他的相关任务还包括识别人物的社会关系网、建立人物的事件信息表、识别人物职位的变化、识别人物处所的变化、识别人物态度立场的变化、识别人物生活状况的变化、识别人物知名度的变化等。

大数据环境下的人物追踪要解决的人物相关信息的同一性判断问题、数据融合问题和跟踪问题，将为与人物处于同一抽象层次的其他对象，如事件、地点、机构等的挖掘提供技术上的参考，也将为抽象对象挖掘本身凝练理论和方法。除学术上的意义外，人物追踪的研究还可以产生一些直接的应用，如人物搜索引擎、对特定人物的跟踪、流行人物发现等。

10.1.2 人物追踪及其处理流程

人物追踪需要多方面的自然语言处理、信息检索以及模式推理技术的支撑，包括文本分类、聚类、信息抽取、模式推理等。人物追踪可大致分为数据预处理、人物模型的同一性判别与动态填充、人物事件信息的组织和若干细粒度挖掘任务的处理。各部分的关系如图 10.1 所示。

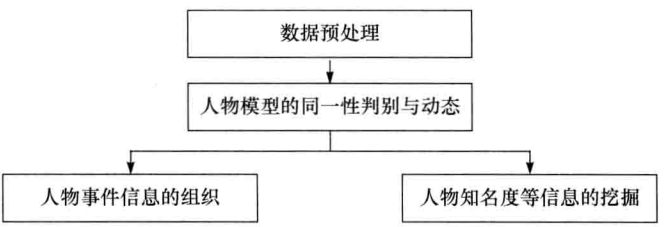


图 10.1 人物追踪的处理流程

10.2 数据预处理

人物追踪可以利用的数据来源很多，如广播、电视、短信、报纸、杂志、网页等，各种形式数据的质量参差不齐，人物信息往往隐含在这些不同来源的大数据中，数据格式多种多样，每种数据的结构也可能不断发生变化。如何对异构多变的数据进行清理，把杂乱、充满噪声的数据转变为适合人物追踪要求的统一数据格式，是数据预处理阶段要完成的主要工作。

本章选取互联网上格式和内容都相对规整的新闻网页作为研究对象。与其他数据源相比，新闻网页具有格式规范、权威性高、报道及时等特点，因此成为人物追踪研究的理想数据源。

10.2.1　数据预处理的主要流程

人物模型中的信息主要由人物名称、基本属性信息、介绍性信息和普通词场等部分构成。数据预处理应能够从含有噪声的数据中挖掘出上述相关的信息。如果数据来源于互联网的新闻网页，则这个阶段涉及的内容主要包括网页正文提取与噪声过滤、人名识别、人名指代、人物对应语段的确定、时间和时序标签的确定、人物属性的抽取等。网页噪声过滤主要解决的是从网页中提取出正文、标题和报道时间等信息，过滤掉广告、链接等噪声；人名识别主要是对内容进行汉语词法分析，识别出人名标签，尤其需要进行人名/非人名的区分；在人名指代处理阶段，主要是识别出人名的指代词，把潜在词场转化为现实词场；确定在一篇报道内跟某一人物相关的语段范围，确定该人物对应的词场；确定这些语段所描述的事件所发生的具体时间，或与其他事件相比的时间顺序；识别人物的属性信息，包括基本属性信息和介绍性信息，并填充进人物模型，形成人物模型片段。

10.2.2　网页正文提取与噪声过滤

对于人物追踪来说，除了报道正文和标题等信息以外，其他所有内容均可视为噪声。这里需要从新闻网页中提取出新闻的正文和标题以及报道时间。简单的处理方法可以是抽取 HTML 页面中“TITLE”标记所夹的内容作为标题，但由于 TITLE 内容往往很随意，不一定能够概括报道的真实内容。为了得到高质量的新闻标题，只能分析挖掘蕴涵在网页文字中的标题。这里重点研究针对汉语的新闻网页中如何提取新闻正文、新闻标题和报道时间的方法。

1. 网页解析算法

网页大多是基于 HTML 语言写成的，网页噪声过滤多通过对 HTML 页面的解析来完成。从 HTML 页面解析的基本技术手段来看，如果处理的对象是特定领域的格式较为简单、规整的网页，利用 HTML 标记分布规律和 HTML 标记间的关系，效果较好。如果处理的是样式结构较为复杂、格式缺乏共性的网页，利用页面视觉特征或 TABLE 标记的布局特性比较有效，但二者各有利弊：视觉特征很好地反映了页面的外在样式与页面内各语义块间的内在联系；TABLE 标记的布局特性则从页面的结构组织上反映了页面的结构与各语义块间的联系。前者分析精细但知识表示模糊，后者规则明确但缺乏微观把握。

为了有效处理新闻网站上的网页，可以把这两种方法综合起来。这里提出综合利用 TABLE 标记和视觉特征的页面解析(table and vision based page segmentation，TVPS)算法。页面解析算法能够将网页内各个语义块进行划分并对各个块进行分析，判别其是属于索引块还是内容块；针对内容块，还可进行精细挖掘，抽

取出标题和报道时间等信息。在综合分析页面中各个块的基础上，判断当前页面是索引页还是内容页，然后把索引页和索引块中的信息视为噪声过滤掉。页面解析算法对 HTML 文档结构分析如图 10.2 所示。算法主要步骤如下。

(1) 构造 HTML 标记树

首先对 HTML 代码进行预处理，补全“”、“<hr>”等结束标志，去除一些无用标记，如“form”、“img”、注释标记等，构造节点树。构造 HTML 标记树可通过编写递归算法来实现，树中每个节点保留字体大小、粗细、颜色、背景等信息，父节点的属性可以继承到子节点上。为了能够有效地对 HTML 文件中的语法错误进行处理，采用 W3C 提供的 HTML 词法分析器进行 HTML 标志名和标志属性的提取。该词法分析器可以分为初始状态、标记开始状态、文本状态、标记结束状态、结束状态等，每个状态分别调用相应的触发函数。

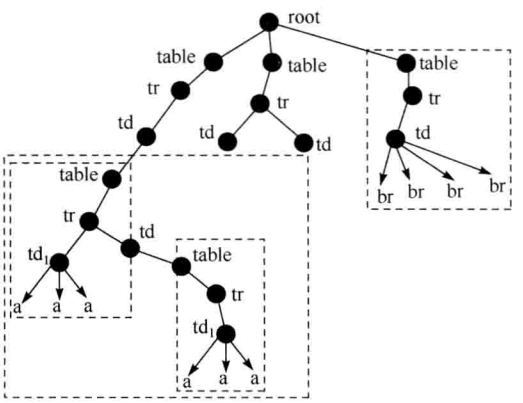


图 10.2 TVPS 算法分析 HTML 文档结构示意图

(2) 对无用子树进行剪枝

对记录信息进行识别，如果叶子节点不是文本或链接的子树，即不含记录信息，则它们被认为是无用子树，应该把这样的子树剪掉。

(3) 分块并寻找各块标题

首先把各底层的 Table 标记对应的子树合并成一个块，把该块中大多数记录的视觉属性作为该节点的视觉属性。如果发现最左端那条记录比其他记录醒目，则把该条记录定为标题。其中，记录 A 是否比记录 B 醒目可采用文献(李晓黎，2001)提出的五个判断因素：A 的字是否比 B 的大；A、B 的颜色是否不同；A、B 的字体是否不同；A 是否是黑体；A 的周围是否存在空行等因素判断。当有两个或两个以上的因素为真时，“A 比 B 醒目”的命题为真。另外，因为有的语义块标题和记录内容分别位于页面内不同的 Table 中，所以有必要对剪枝和合并后的子树进行检查。

(4) 判断各语义块及页面类型

在各块内部统计链接文字与非链接文字的比例，如果该比例大于设定的阈值，则认为该块是索引块，否则是内容块。如果一个页面中全部语义块都为索引块，则该页面为索引页，否则为内容页。

(5) 识别内容页的报道时间

为了有效识别出内容页报道的时间，可采用位置和词性双重约束的方式。考虑内容块标题和正文之间的文字，判断它们的词性，对词性为“数词(m)”或“时间词(t)”的文字串，将其挖掘出来并作为报道时间。采用词法分析器 ICTCLAS 进行词性判断。

2. 实验结果与分析

我们收集了 2000 余篇汉语新闻网页，收集网页采用的是元搜索方式，即制作一个元搜索采集器，输入一些起始关键词。该采集器就能自动从“百度”新闻搜索引擎中采集下来关键词对应的查询结果。在这里输入的关键词是 5 个中文人名，采集到的初始网页一共 2632 篇，经过网页去重，剩下 2110 篇。遗憾的是，通过新闻搜索引擎采集下来的网页都是内容页面。为了有效验证算法针对索引页与内容页划分的性能，又人为地加进了 40 篇索引页面。这样数据集合一共有 2150 篇网页。

需要测试的内容包括索引页与内容页的划分、新闻正文的识别、网页噪声的过滤、报道时间识别、正文标题识别。新闻正文识别和网页噪声过滤是否正确的判断标准比较模糊。我们规定仅当新闻正文的首段和尾段的定位偏差都不超过两个自然段时，才认为新闻正文识别正确；仅当一篇网页上的噪声过滤掉 95%以上时，才认为噪声过滤正确。这样，设某一项正确识别的结果数为 m，需要识别的样本数为 n，则采用准确率 P 来衡量每一项的效果，即

$$P=m/n$$

使用页面解析算法对数据集中的网页进行处理，实验结果还是令人满意的。由于网页解析处于整个系统的前期阶段，它的性能将直接影响到后续的处理。从表 10.1 的结果来看，页面解析算法基本可以达到实用效果。

表 10.1　页面解析算法的实验结果

测试项	准确率/%
索引页过滤	99.0
新闻正文识别	93.6
网页噪声过滤	89.5
报道时间识别	86.2

续表

测试项	准确率/%
正文标题识别	78.6

10.2.3 人名识别

从计算机角度来看，文本只是字符的集合。字符集合并没有对人物的信息加以特殊标记。为了有效地挖掘人物的信息，有必要识别人物信息的触发特征。人名作为人物标识，自然成为触发特征的首选。

人名识别是指把人名从无分隔符号标记的字串中准确自动识别出来的过程。英语中人名的识别相对容易，但对中文人名识别来说比较困难，因为中国人名的构成具有多样性，人名内部往往相互成词，人名与其上下文经常组合成词、同源歧义的冲突等。另外，由于标准不统一，外文人名翻译还常常产生差异和混乱。

针对中国人名的自动识别问题，人们已经做过探索并提出了多种解决方案。根据其使用的方法不同，这些方案大致可分为规则方法、统计方法，以及规则与统计相结合的方法。规则方法简单且有针对性，当缺乏大规模语料的时候比较适用；统计方法需要大规模标注语料，所用方法一般比较复杂。目前的研究一般倾向于将规则和统计的方法结合起来，根据调整二者的比重来满足具体情况的需求。

在人名识别方面，可直接使用我们的汉语词法分析系统 ICTCLAS，它采用基于角色标注的中国人名自动识别方法。该方法主要采用隐马尔可夫模型对分词结果进行人名构成角色的标注，然后根据各个不同的角色，进行简单的模式匹配并最终识别出人名。在 ICTCLAS 中，首先在低层的 HMM 识别过程中识别出普通不嵌套的未登录词，然后在此基础上，利用相同的方法采取高层隐马尔可夫模型，通过角色标注计算出最优的角色序列，从而进一步识别出嵌套的未登录词。例如，切分片段“周/恩/来/和/邓/颖/超/纪念馆”，ICTCLAS 先识别出“周恩来”和“邓颖超”为人名 PER，得到新的词类序列“PER/和/PER/纪念馆”，最终识别出该片段为机构名而不是人名。

采用 ICTCLAS 对新闻内容进行词法分析，并把词性标签为“nr”、“nrf”的词语视为人物的姓名，作为人物信息挖掘的触发词汇。

10.2.4 人名指代处理

文本中对人物的标识词汇不一定是人物姓名，还可能是人名的指代词。具有指代作用的词主要包括姓、名、人称代词“他/她”等。只有把这些指代词跟人名的对应关系整理清楚，才能找全人物信息的描述范围。指代识别正确了，潜在的词场才能转化为现实的词场，才能丰富人物实体的特征数据。

人名指代处理中比较复杂的是人称代词的消解，即确定人称代词与先行词之间的关系，从而明确人称代词究竟指代什么对象。人称代词的消解是自然语言处理中十分重要的问题。针对汉语篇章，在必须解决人称指代问题时，一般都遵循“最近匹配原则”。指示词与先行词之间可以跨越多个句子，距离越大，先行词候选集规模可能越大，解决消解问题的复杂性也就越大。我们的做法没有涉及人称代词的指代消解，而是只对名或姓的指代进行了处理，如“王建民”先行词与“王校长”、“建民同志”之间的指代关系，实现了一个基于规则的人名指代消解系统，只有具有较高准确率的规则被应用进来，只处理“姓”（词性标注为“nr1”）和“名”（词性标注为“nr2”）的指代词，且严格遵循先行词中“姓”和“名”部分的完全匹配。

10.2.5　人物对应语段的确定

在一篇新闻报道中，尤其是新闻报道比较长的时候，跟一个人物相关的语段范围是有限的，它往往不是全文，可能只是报道的一部分。如果要精确地确定人物在报道中的作用语段，需要复杂的篇章分析。在人物追踪中，挖掘人物在报道中相关的语段，可以利用一些启发式规则，以人名及其指代词为中心来对文档进行分割。这些启发式知识主要是利用人名及其指代词在文档中的位置分布来确定人物的作用范围，确定的规则如下。

① 如果人名或其指代词出现在报道的标题和正文首段，则该人物的作用范围是报道全篇。

② 如果人名或其指代词出现在报道的其他段落，则该人物的作用范围是当前段落。

③ 如果某一段落（尾段除外）相邻的两个语段都是某人物的有效段落，则这个段落也是该人物的有效段落。

如果一篇报道描述了 N 个人，就对该报道拆分重组成 N 个有效描述的集合 $\{P_1, P_2, \cdots, P_N\}$ 分别与这 N 个人相对应。采用上述规则来确定人物对应的语段，一个简单的例子如图 10.3 所示。

继续对日本进行正式友好访问 吴邦国委员长抵达大阪

本报大阪（日本）9 月 9 日电 记者史宗星报道：全国人大常委会委员长吴邦国 9 日抵达日本第二大城市大阪，继续对日本进行正式友好访问。

下午，吴邦国考察了松下电器公司。松下电器公司会长森下洋一、社长中村邦夫向吴邦国介绍了松下的产业发展和与中国合作的情况。松下公司是最早进入中国市场的日本公司之一，在中国开办了 53 家从事生产、研发或销售的独资或合资企业，直接投资达 70 亿元。中国已经成为松下海外事业发展的重要支柱。松下已经制定 2005 年中国市场年销售达 1 万亿日元（约 700 亿元人民币）的目标。

吴邦国积极评价了松下同中国的合作，表示很高兴地看到松下在同中国合作的过程中自己也得到了壮大，已经成为中国消费者家喻户晓的日本公司。

他表示，中国和日本现在又面临着双边关系发展的新机遇，希望有远见的日本公司能充分分享中国高速增长的市场，一起来创造更美好的明天。

傍晚，吴邦国会晤了大阪府知事太田房江、大阪市市长矶村隆文和日本关西经济界人士。

图 10.3 新闻报道正文示例

上述新闻报道中出现“吴邦国”、“史宗星”、“森下洋一”、“中村邦夫”、“太田房江”、“矶村隆文”。其中“吴邦国”在报道的标题中出现，则全篇报道都认定与他有关，所以“吴邦国”的作用范围是整个文档；“史宗星”是报道记者名，可通过简单的规则过滤掉；“森下洋一”和“中村邦夫”只在文中第二段出现，所以把第二段分别作为这两人的作用范围；“太田房江”和“矶村隆文”的作用范围是报道的最后一段。这样可以把这篇报道重组成 3 个描述范围，分给不同的 5 个人。

10.2.6 时间和时序标签的确定

时间是事件描述的一个重要因素，是区分不同事件的重要依据。为了表明一篇新闻报道发布的时间，一般报道本身会有一个时间戳记。根据描述的时间粒度的不同，可以把时间表达式划分为日期表达式和时刻表达式。从字面上看，同一个时间表达式可以有不同的表达方式，如“2013 年 3 月 1 日”与“二零一三年三月一日”。虽然人工很容易判断是同一个日期，但在计算机看来，却是完全不同的两个字串，所以有必要对时间表达式进行识别和规范化处理。

另一方面，由于新闻报道本身一般会有一个时间戳记，所以在新闻报道中经常使用相对时间表达式，如“下午”、“3 日”、“明年”等。为了更好地确定人物对应描述所发生的时间，有必要对时间表达式识别并规范化处理之后进行一定的时制运算。这里对时制运算不作进一步的研究，只是简单地把新闻报道本身的时间戳记作为人物对应描述发生的时间，并只对典型的几种时间描述格式进行规范化处理。

10.3 人物属性抽取

人物的基本属性信息主要包括性别、民族、生日、家乡、亲属等。人物的介绍性信息主要包括职位、住址、配偶、同事、联系方式等。与普通词场相比，这些信息对人物的刻画更加深入和精确。如何从文本中得到这些信息，也需要更加精确有效的手段。其中，信息抽取尤其是实体关系抽取是解决这一问题的有效方法。

10.3.1　人物属性抽取的总体框架

人物属性抽取可以看做是实体关系抽取的具体应用。实体关系抽取一直是信息抽取研究领域的重要课题，得到相关研究者的重视。MUC-7 和 ACE 评测会议都把实体关系抽取列为重要的评测内容。其中，ACE 评测提供的语料不仅是英文，还包括中文和阿拉伯文。与自然语言处理领域的许多问题类似，实体关系抽取最初也是使用知识库的方法，这种方法需要经验丰富的领域专家手工构筑大规模的知识库（主要是编写抽取模板），需要大量的人工劳动，而且用这种方法实现的实体关系抽取系统很难从一个领域移植到另一个领域。为了应对这种挑战，近年来研究者倾向于把基于知识库的方法改换成机器学习的方法。这种方法往往把实体关系抽取看作是一个分类问题，一般来说它分为训练阶段和测试阶段。训练阶段需要人工标注一定规模的语料库，主要是标注出存在着特定关系的实体对，然后系统从标注好的语料中学习并构造出分类器。在测试阶段，系统对具有潜在关系的实体对使用分类器进行分类，并决定该实体对是否真正存在着特定的关系。

多种机器学习的方法都可以有效应用在实体关系抽取中，采用分类思想进行实体关系抽取属于监督式机器学习方法。为了减少语料的标注量，有的研究者引进了弱监督式机器学习（Qiu et al.，2011）。按照分类器的表示形式划分，实体关系抽取主要有特征向量表示法和 Kernel 表示法。特征向量表示法主要是把对象实例数值化成向量形式，然后对于给定的一组训练数据$(x_1, y_1)(x_2, y_2)\cdots(x_n, y_n)$，其中$y_i \in \{-1, +1\}$，学习一个分类函数 f，使得对于给定的新的特征向量 x'，f 能够将其正确地分类，即 $f(x') = y'$。与特征向量学习方法不同，Kernel 表示法不需要构造特征向量，而是直接使用实例的原始形式作为处理对象，需要计算的是任何两个对象之间的 Kernel 函数。Kernel 表示法的优点是可以搜索更广阔的特征空间，因此往往能够达到比特征向量法更佳的效果。当处理大规模数据集时，Kernel 表示法的速度要比特征向量法慢。

本章关注的是大规模互联网真实环境下人物属性的抽取。与其他抽取任务相比，所面临的问题有以下三个特点，即大数据、人名和各个属性之间表达方式灵活多样（简单的模板定制方法很难把各种情况概括全面）、抽取出来的属性信息不仅用于结果呈现，还会参与后续的计算。基于以上三个特点，采用监督式机器学习的方法来抽取人物的属性，即手工标注一定规模的语料，然后训练出分类器用来判断新的实体对是否存在搭配关系。分类器采用特征向量的表示方法。抽取方法的主要流程如图 10.4 所示。

10.3.2　标注人物属性抽取语料

本章关注的是新闻网页中人物属性的抽取问题。遗憾的是，尽管实体关

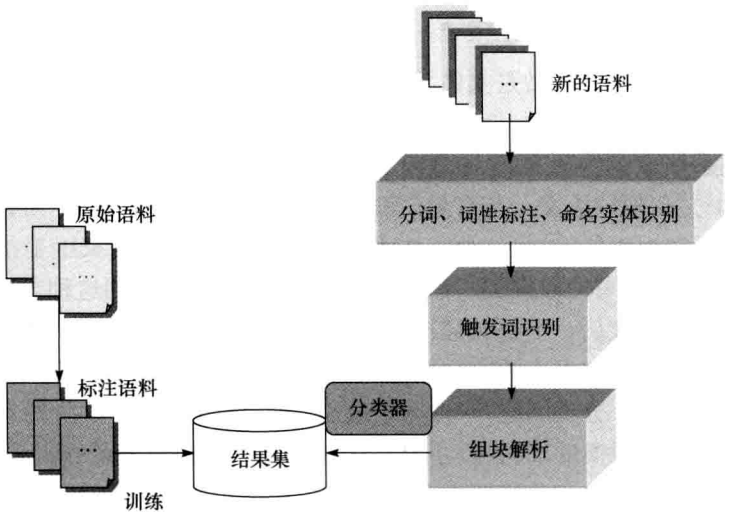


图 10.4 人物属性抽取方法流程图

系抽取的研究很多，但还没有找到符合本章特点要求的语料。可以收集新闻网页并进行标注，建立一个汉语的、针对新闻网页的人物属性抽取语料集。这里网页的收集仍然采用元搜索的方式，采集下来的原始网页 7731 篇，经过进一步的噪声过滤（把新闻正文中不含输入人名的网页过滤掉）和网页去重，剩下 4315 篇。在新闻网页中出现的人物属性，比较典型的有七类，即性别、民族、生日、籍贯、职位、住址、家庭关系。本章对 4315 篇新闻正文中起始输入的 17 个人名的属性进行标注。为了使手工标注的词语边界与机器自动分析的相一致，属性标注是在词法分析的结果上进行的（采用的词法分析软件是 ICTCLAS）。语料标注的格式示例如下。

（1）人物职位语料

例 1：[东安县职业中专/nt 校长/n][李华/nr]被/pbei 授予/v[永州市/ns 优秀/a 教育工作者/n]荣誉/n 称号/n

例 2：[李华/nr]9/m 次/qv 被/pbei 评为/v[县/n 先进/a 教育工作者/n]

（2）人物住址语料

例 1：[李华/nr]，/wd 家/n 住/vi[北京市/ns 海淀区/ns]

例 2：一个/mq “/wyz[濮阳县城关镇/nt 南环/ns 西路/ns 163/m 号/q][李华/nr]”/wyy 的/udel 汇款/n 地址/n

(3) 人物亲属关系语料

例 1:姐妹/n 俩/m[李中/nr]和/cc[李华/nr]

例 2:那/rzv 妇女/n 名叫/v[赵桂花/nr],/wd 这天/rzt 打算/v 带/v 着/uzhe 儿子/n[李华/nr]从/p 深圳/ns 乘/v 1506/m 次/qv 列车/n 回/v 江西/ns 老家/n

(4) 人物性别语料

例 1:[李华/nr],/wd[男/b],/wd 住/vi 广西/ns 全州县/ns 全州/n 镇/n 黄中路/ns 95/m 号/q

例 2:[李华/nr][先生/n]被/pbei 任命/v 为/v 东安/nz 四中/n 校长/n

在 4315 篇语料的 17 个人名对应的属性信息里,“性别”信息出现 113 个,“民族”出现 15 个,“家庭关系”出现 23 个,“出生年月”出现 27 个,“家乡”出现 77 个,“住址”出现 25 个,“职位”出现 2301 个。针对每种属性,把语料平均分成 3 份,且保证该属性在每份中所占的比重相同。分别拿其中 2 份做训练,1 份做测试,采用 3 次交叉验证的方式,最后的结果取 3 次结果的平均,由此实现对人物属性抽取语料的标注。

10.3.3　分类器模型

1. 普通词场的表示方法

可以采用特征向量来表示分类器。特征向量的内容来自于围绕人名和属性周边的若干词语和符号,也就是说,把人名和属性看作是一个二元关系＜tag1, tag2＞,围绕 tag1 和 tag2 的词汇按照位置可以划分为 Left、Middle 和 Right 关系,这些信息构成了五元组(Left, tag1, Middle, tag2, Right)。在实际中,Left 取 tag1 左边窗口大小为 2 的词,如果遇到标点符号和句子边界则被截断;Right 取 tag2 右边窗口大小为 2 的词,如果遇到标点符号和句子边界则被截断;Middle 取 tag1 和 tag2 中间的词。Left、Middle、Right 中有好多词汇起不到表征的作用,为此可以建立停用词性表,把 Left、Middle、Right 中的助词、人名、标点符号等忽略词去掉。经过这样的处理,Left、Middle、Right 可能会出现值为空的情况。当 Middle 为空时,表示 tag1 和 tag2 紧邻,其关系可能会更加密切(实践中,可以对空值 Middle 采用特殊符号“&”加以标记)。

采用向量空间模型来表示分类器时,向量中每一维度的权重采用 TF 测度值来表示。为了简化,把每一属性的五元组中 Left、Middle、Right 的词汇混合成一个向量。

2. 基于概念义原的语义计算

如果把互联网中的新闻网页看作是一个大的语料测试集，那么它是大数据规模的。相比之下，所标注的训练集规模就比较小，一般而言很难覆盖互联网上的所有语言情况；另外汉语的表达方式灵活多变，描述同一概念的词汇多种多样，也可能存在着字面上的差异。这种字面描述的差异造成了关键性特征的分散，也使特征向量无法有效起到标识作用。为了解决这个问题，可以引进语义计算的方法。Zhang 等(2005)采用两两词汇进行语义计算然后加权平均的方法来计算两个英语句子的语义相似度。类似地，也可以把该方法引入到两个汉语特征向量的语义相似度计算中来。然而，该方法倾向于引入大量语义噪声，另外英语和汉语的单个词汇的语义计算方法各有不同，在英语中效果好的方法不一定适用于汉语环境。

为此，本章只在分类器训练时的特征向量形成阶段引入语义计算。基本方法是对特征向量普通词场中的词进行语义聚类，把聚类结果的每个词团作为统一的维度，并把该词团中每个词的权重相加作为该维度的权重。

可以借用 HowNet 进行语义计算。由于 HowNet 的基本元素叫义原(指最基本的、不易于再分割的意义的最小单位)，HowNet 总是通过用一系列的义原，利用某种知识描述语言来描述一个概念，因此可以把基于义原体系得到的词语聚类结果称为概念义原。这样，特征向量可以表示为

$\mathrm{SEM}\{(\mathrm{sem}_1(\mathrm{w}_1,\mathrm{w}_2),\mathrm{k}_1),(\mathrm{sem}_2(\mathrm{w}_3,\mathrm{w}_4,\mathrm{w}_5),\mathrm{k}_2),\cdots,(\mathrm{sem}_m(w_{n-2},w_{n-1},w_n),k_m)\}$其中，$\mathrm{sem}_1$，$\mathrm{sem}_2$，…，$\mathrm{sem}_m$ 是概念义原；w_1 和 w_2 是归为 sem_1 的两个词；w_3、w_4 和 w_5 是归为 sem_2 的三个词；k_1 是 sem_1 的权重(w_1 和 w_2 的权重之和)；k_2 是 sem_2 的权重(w_3、w_4 和 w_5 的权重之和)。

概念义原对应的是词在特定上下文中所表达的概念，因而字面上不同的词可能对应相同的概念义原。例如，“死”、“死亡”、“丧生”、“毙命”、“遇难”对应的概念义原均为“die/死”，因为它们在 HowNet 中的定义均为“DEF＝die/死”。这样，特征向量中每个维度都以概念义原作为统计对象而不是以词作为统计对象，可以消除表达同一个概念的不同的词在字面上的差异，突出关键性的特征，解决语料稀疏的问题。

基于概念义原，我们关注如下相似度的计算，即词语间相似度计算、概念义原间相似度计算、概念义原和普通词场相似度计算。

(1) 词语间相似度计算

采用整体的相似度等于部分相似度加权平均的做法。首先将一个整体分解成部分，再将两个整体的各个部分进行组合配对，通过计算每个组合对的相似度的加权平均，得到整体的相似度。对于两个义原的相似度，采用根据上下

位关系计算得到语义距离并进行转换的方法。

对于两个汉语词语 W_1 和 W_2，如果 W_1 有 n 个义项(S_{11}，S_{12}，…，S_{1n})，W_2 有 m 个义项(S_{21}，S_{22}，…，S_{2m})，则 W_1 和 W_2 的相似度是各个概念相似度之最大值，即

$$\mathrm{Sim}(W_1,W_2)=\max_{i=1,2,\cdots,n,j=1,2,\cdots,m}\mathrm{Sim}(S_{1i},S_{2j})$$

(2) 概念义原间相似度计算

需要通过聚类的方法把语义上相同的词语聚在一起。在凝聚式层次聚类过程中，需要衡量两个概念义原的相似度以便决定是否需要进行合并。采用常用的 Average Link 方法来计算两个概念义原的相似度，即

$$\mathrm{Sim}(\mathrm{Sem}_p,\mathrm{Sem}_q)=\frac{\sum_{i=1}^{m}\sum_{j=1}^{n}\mathrm{Sim}(W_{pi},W_{qj})}{mn}$$

其中，Sem_p 和 Sem_q 分别是两个概念义原；W_{pi} 和 W_{qj} 分别是概念义原 Sem_p 和 Sem_q 中的词汇。

(3) 概念义原和普通词场相似度计算

概念义原和普通词场的相似度计算发生在分类器的测试阶段，即把分类器本身的特征向量采用概念义原来表示，但把测试文档采用普通词场向量来表示。基于标准的余弦公式，它们之间的相似度为

$$\mathrm{Sim}(\mathrm{Sem}_r,S_i)=\cos(\mathrm{Sem}_r,S_i)=\frac{|\mathrm{Sem}_r S_i|}{|\mathrm{Sem}_r||S_i|}$$

当计算(Sem_r，S_i)时，将 Sem_r 和 S_i 中出现相同词语的维度进行对应，但在计算二者权重乘积时，把 Sem_r 中这个词语的权重用它所属的概念义原的权重来代替，这就使得 Sem_r 和 S_i 之间的相似度从概念的意义上得到了扩大。

3. 训练分类器

分类器采用简单高效的中心分类法，训练与分类速度很快，因此在文本分类尤其是大数据分类中得到了广泛的应用。这样就从训练语料中训练出性别、民族、出生年月、籍贯、职位、住址、家庭关系等中心向量和各自最佳的阈值以及数据集中人名和属性所间隔的最大窗口。采用“和中心”来表示中心向量，即 $\vec{C_i}=\sum_{d\in C_i}\vec{d}$，决策规则为

$$C=\arg\max_{C_i}(\mathrm{sim}(\vec{d},\vec{C_i}))=\arg\max_{C_i}(\vec{d}\cdot\vec{C_i})$$

选取训练样本中，离中心最远的距离作为评判测试样本的阈值；选取训练样本中人名与各个属性所间隔的最大词数作为该属性的窗口。人物属性训练语料中 tag1 和 tag2 所间隔的最大窗口长度如表 10.2 所示。

表 10.2 人物属性项对应的窗口长度

人物属性项	窗口长度
性别	4
民族	4
出生年月	6
籍贯	8
职位	45
住址	8
家庭关系	12

4. 测试分类器

测试阶段的步骤主要包括词法分析、触发词识别、组块解析和分类四个步骤，其中词法分析可以调用词法分析软件 ICTCLAS，下面主要介绍后三个步骤。

(1) 触发词识别与触发词库的建立

为了从未经标注的语料中发现人物属性，可以建立触发词库。为每种属性建立触发词库的方法如表 10.3 所示。

表 10.3 人物属性触发词库的建立方法

人物属性项	触发词库
性别	HowNet 里标注为“human\|人，male\|男”或“human\|人，female\|女”的词语，且去掉了一些特别宽泛易混淆的词，如“生”等
民族	全国 56 个民族的名称
出生年月	词性为“t”，且以数字与“年”、“月”组成的词语
籍贯	词性为“ns”的词语
职位	Hownet 里标注为“human\|人，# occupation\|职位”的词语，且去掉了一些特别宽泛易混淆的词，如“人”等
住址	词性为“ns”的词语
家庭关系	词性为“nr”的词语

对于一个新的句子，先对其进行分词、词性标注和命名实体识别，然后扫描该句子，如果发现满足某触发条件的词语，则开始判断该属性是否是特定人物的属性。

(2) 基于规则的组块解析

组块解析是自然语言处理里面一项重要的基础性研究，它对于句法分析、信息

检索、信息抽取和机器翻译都有重要的实践意义。这里主要关注的是把人物属性触发词对应的最大名词短语(MNP)识别出来。限于篇幅,这里不对最大名词短语识别进行深入研究,只是在词性标注的结果上利用简单的规则来识别人物属性对应的最大短语,方法也是基于自然语言中存在的长尾综合效应,即绝大多数事例采用经常出现的、非常少量的表达方式来描述。表 10.4 是识别人物属性短语所采用的规则。另外,对并列的职位短语和人名短语进行了识别,即如果两个职位短语或两个人名之间的间隔词为“、”或“和”,则它们之间属于并列关系,并列的短语共享相同的特征向量。

表 10.4　人物属性短语识别规则表

人物属性项	规则方法
性别	仅触发词即可
民族	仅触发词即可
出生年月	从触发词往后顺序找词性为“t”且中心为“月”、“日”的词
籍贯	从触发词往后顺序找词性为“ns”、“f”、“a”、“n”、“b”、“s”的词,遇到“人”结束
职位	从触发词往前顺序找词性为“n”、“nt”、“ns”、“nz”、“nl”、“ng”、“b”、“f”、“vn”、“m”、“q”、“s”、“nsf”、“ag”、“a”的词
住址	从触发词往后顺序找词性为“ns”、“f”、“a”、“ng”、“m”、“q”、“n”、“b”、“s”的词,遇到“人”结束
家庭关系	仅触发词即可

(3) 分类

在找到满足触发条件的词语,并识别出最大的属性短语 Attribute_i 以及与该短语处于并列地位的其他短语后,在允许的窗口 Window_i 范围内,寻找出现的人名。对于任一属性短语 Attribute_i,如果属性左边 Window_i 窗口内存在人名,则可以挖掘出对应的特征向量 $\text{Vector}_{\text{Left}i}$;如果属性右边 Window_i 窗口内存在着人名,则可以挖掘出对应的特征向量 $\text{Vector}_{\text{Right}i}$。特征向量的构造方法同“普通词场的表示方法”中的方法,然后与训练好的分类器中的语义向量进行相似度运算。这样就可以计算出该属性短语对应的左侧向量的相似度 LSim 和右侧向量的相似度 RSim,LSim 与 RSim 哪个大,如果超过设定的阈值,该属性就属于哪侧的人名(包括并列的人名)。

5. 实验结果及分析

在测试过程中我们更关注高的准确率，因为在网页数量趋向于无限大的情况下，准确率比召回率更重要，因此在计算 F 值时，我们认为准确率的重要性是召回率的 2 倍，即 $F=\frac{3PR}{P+2R}$。

表 10.5 和表 10.6 分别是采用普通词场向量的人物属性抽取效果和采用概念义原向量的人物属性抽取效果。图 10.5 是效果对比图。

表 10.5 采用普通词场向量的人物属性抽取效果

人物属性	阈值	答案总数	结果总数	正确数	准确率	召回率	F 值
性别	0.05	113	113	108	0.955752	0.955752	0.955752
民族	0.10	15	15	15	1.000000	1.000000	1.000000
出生年月	0.20	27	27	27	1.000000	1.000000	1.000000
籍贯	0.40	77	74	65	0.878378	0.844156	0.866667
职位	0.30	2301	1998	1973	0.987487	0.857453	0.939971
住址	0.35	25	13	13	1.000000	0.520000	0.764706
家庭关系	0.28	38	24	14	0.583333	0.368421	0.488372

表 10.6 采用概念义原向量的人物属性抽取效果

人物属性	阈值	答案总数	结果总数	正确数	准确率	召回率	F 值
性别	0.16	113	110	108	0.981818	0.955752	0.972973
民族	0.10	15	14	14	1.000000	0.933333	0.976744
出生年月	0.20	27	27	27	1.000000	1.000000	1.000000
籍贯	0.46	77	65	62	0.953846	0.805195	0.898551
职位	0.35	2301	1998	1973	0.987487	0.857453	0.939971
住址	0.41	25	14	14	1.000000	0.560000	0.792453
家庭关系	0.47	38	26	25	0.961538	0.657895	0.833333

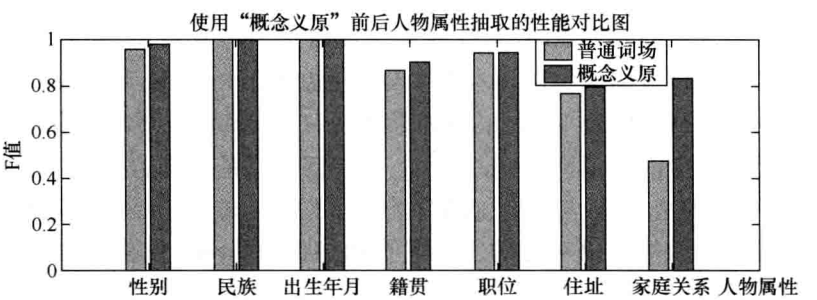


图 10.5 效果对比图

从上面结果可以看出，采用的人物属性抽取方法取得了比较好的效果。跟前人工作相比，我们取得的实体关系抽取性能较高，原因是所使用的语料不同。这里使用的语料是严格按照互联网上新闻网页中各属性模式出现的频率来建立的，实验结果反映了采用该方法在实际互联网环境下，抽取人物属性所能取得的效果。

采用基于概念义原的特征向量后，各项属性抽取的结果都有所提高。其中，家庭关系抽取的性能提高得最为明显，因为家庭关系的特征词最为分散（如“弟弟”、“父亲”、“前夫”、“儿子”、“妻子”、“侄女”“夫妇”、“姐妹”等），只有通过概念义原才能把它们统一到一起，突出关键性的特征。按照人物各个属性的结果逐个分析，可以看到。

① “性别”、“民族”、“出生年月”达到了很高的性能，是因为这两种属性的表述方式比较固定。

② “籍贯”、“职位”、“住址”、“家庭关系”可以达到较高的准确率，召回率略低，这是因为这四个属性的描述方式比较灵活，个别的例子系统没有识别出来。拿“职位”来说，因为在真实的新闻网页中，出现的职位短语绝大多数是与人名紧邻的，能够识别正确的职位基本上是跟“人名”紧邻的“职位短语”。不紧邻的情况识别得不够好，例如“[张林/nr]今年/t 27/m 岁/qt ，/wd 原/d 是/vshi 上海/ns 一/m 家/q [外企/n 白领/b]”，“聘任/v[张林/nr]先生/n 为/p[公司/n 财务/n 总监/n]”，这些句子中“职位词”与“人名”离得太远，与职位密切相关的动词性特征，如“是/vshi”、“聘任/v”等淹没在其他噪声词汇中，没有起到关键性的作用。极少数识别错误的例子，例如“面对/v 百/m 位/q 民营/b 企业家/n ，/wd[张高丽/nr]讲/v 了/ule 省委/n 、/wn 省政府/n 的/udel 期待/vn”——把“民营/b 企业家/n”看作“张高丽/nr”的职位；再如，“小/a 歌手/n[陆放/nr]和/cc[张艺谋/nr]申/vg 博/ag 片/ng”——把“小/a 歌手/n”识别成了“张艺谋/nr”的职位，因为“小歌手”是“陆放”的职位，“张艺谋”和“陆放”识别成了并列人名短语。如何解决上述问题，也是下一步的研究内容之一。

10.4　本章小结

本章介绍了针对汉语的新闻网页人物追踪数据预处理的主要流程。数据预处理主要解决的是把蕴涵在文本中的人物相关信息挖掘出来，建立起人物模型的片段，为后续的处理做准备，其内容主要包括网页噪声过滤、人名识别、人名指代处理、人物对应语段确定、时间和时序标签确定、人物属性抽取。在网页噪声过滤阶段，提出了综合利用 TABLE 标记和视觉特征对页面进行解析的算法 TVPS，并用该算法从新闻网页中提取出新闻报道的正文、标题和报道时间等信息，过滤掉广告、链接等噪声。人名识别时，使用 ICTCLAS，把词性标签为“nr”、“nrf”的词语视

为人物的姓名，作为人物信息挖掘的触发词汇。采用了若干简单有效的规则进行人名指代的处理、人物对应语段的挖掘和时间标签的确定。采用监督式机器学习的方法来抽取人物的属性，即手工标注一定规模的语料，然后训练出分类器用来判断新的实体对是否存在搭配关系。为了消除描述同一概念的词汇在字面上的差异，提出了基于概念义原的特征向量表示形式，实验证明该方法能有效提高人物属性抽取的效果。

相关工作与扩展阅读

人物角度的信息组织始于自然语言处理方面的研究，主要集中在人名消歧。人名的消歧可以看做是自然语言中共指消歧的一种类型，即识别出人名和它的指代词之间的对应关系。随着信息处理的不断加深，人名消歧也由单文档消歧扩展到跨文档消歧方面。近年来，随着信息检索技术的快速发展，一些搜索引擎采用对搜索结果进行聚类的方式来减少结果中的歧义现象。由于这类搜索技术意在对通用领域信息进行处理，对解决人名歧义的问题效果就不够理想。为此，一些研究者专门针对 Web 搜索结果中人名消歧的问题进行了研究(Al-Kamha et al.，2004；Wan et al.，2005)。与普通文本相比，Web 搜索结果中的语料有其独特的特点，可以利用的信息也更加丰富。例如，不仅可以利用网页中普通词汇的特征，还可利用在文字中隐含的人物属性的特征，以及利用网页之间的链接关系等。相关工作中，梁浩等(2009a)提出一个基于本体技术并充分利用实例信息的表单属性自动抽取方法，在这一方法中使用 WordNet 来丰富抽取出的关键属性并发现表单中属性间的语义关系。在属性抽取过程中，每个属性被拓展生成一个备选属性集并且以树型数据结构存储，而且备选属性树可以有效地描述属性间的语义关系。实验证明，这一框架结构可以自动抽取 Deep Web 曲表单属性并有效地生成查询条件。梁浩等(2009b)中为提高属性抽取的准确率且实现在语义层面上对查询接口的解读，提出一种以查询接口启发式信息为基础的属性抽取方法，通过使用本体工具对属性集合进行拓展并获取语义描述。在实际的电子商务网站上进行的广泛实验证明了提出方法的可行性与有效性。王宇等(2010)发现网页信息抽取是互联网挖掘的重要课题。为了自动化抽取过程，研究利用特定领域的特征，通过机器学习方法对信息抽取过程进行统一建模，但是对领域特征的依赖使得这类方法难以推广到其他领域中去。因此，对信息抽取问题进行了分析，从中分离出一个可以完全自动化的信息抽取子任务，即有名属性抽取任务。在多个领域的数据集上进行的统计表明这个子任务覆盖了 60%以上的待抽取属性，因此在整个信息抽取中占有重要地位。同时，给出了一种基于扩展领域模型的有名属性抽取方法，实验结果表明这种方法的准确率接近或大于 80%，召回率大于 90%。

参 考 文 献

李晓黎．2001．WEB 信息检索与分类中的数据采掘研究．中国科学院计算技术研究所博士学位论文．

梁浩，左万利，任斐，等．2009a．基于本体实例信息的深度网表单属性自动抽取．小型微型计算机系统，30(5)：883-886.

梁浩，左万利，任斐，等．2009b．基于启发式信息的 Deep Web 查询接口属性抽取．计算机研究与发展，46(z2)：423-429.

王宇，谭松波，廖祥文，等．2010．基于扩展领域模型的有名属性抽取．计算机研究与发展，47(9)：1567-1573.

Al-Kamha R，Embley D W．2004．Grouping search engine returned citations for person name queries// Proceedings of WIDM'04.

Qiu J，Zhu L H，Hao J K．2011．Relation extraction based on multi-view ensemble algorithm. ICIC Express Letters，4：1051.

Spink A，Jansen B J，Pedersen J．2004．Searching for People on Web Search Engines. Journal of Documentation，60(3)：266-278.

Taffet M D．2004．Person resolution：resolving multi-referent and multi-morphic person names within and across full text documents. Syracuse University Phd Dissertation.

Wan X J，Gao J F，et al．2005．Person resolution in person search results：webhawk// Proceedings of the ACM 14th Conference on Information and Knowledge Management.

Zhang H P，Sun J，Wang B，et al．2005．Computation on sentence semantic distance for novelty detection. Chinese Journal of Computer Science and Technology，20(3)：331-337.

第 11 章　人物模型组织与基于事件的信息处理

上一章介绍的人物追踪数据预处理流程，只是识别出若干人物相关的零散信息，为人物模型的计算做好了准备，但还没有形成完整的人物模型。我们认为，只有效解决人物的同一性判断问题和数据融合问题，才能凸显出人物实体，进而为人物跟踪打下良好基础。本章将对人物对象挖掘的同一性判断问题和数据融合问题进行论述。人物模型的组织，即同一性判别和动态填充为人物追踪打下基础。基于事件的信息组织能有效组织人物参与的相关事件，能够建立出人物的重大事件信息表和活动报道专集。

11.1　概　　述

在文本空间，人物是比文字对象高一个层次的抽象实体。作为处于抽象层次的实体，它与文字对象所描述的具体名称并不形成一一对应关系。也就是说，可能存在着大量的同名同姓却不同人物的现象，即使是同一人物，也往往有不同的称谓。在数据预处理阶段，形成的只是人物模型的片段，还需要对这些片段性的信息进行同一性判别，即把描述不同人物的模型片段区分开。另外，数据预处理阶段形成的人物模型片段信息是分散的、孤立的，不能代表人物全貌，更无法凸显出人物抽象实体。这就要求不仅要把代表不同人物的模型片段区分开，还要对同一人物的模型片段信息进行有机整合，进而形成完整的人物模型，清晰地呈现人物的履历等信息。为了取得完整的信息，可能需要参照多个异构来源的大数据信息，才能补齐人物的相关信息，因此需要把多个人物模型片段的信息融合在一起，这就涉及大数据信息的融合问题。

然而，在文本提供的人物相关信息有限的情况下，如何有效地对人物进行同一性判别，并不是一件容易的事。一般来说，需要对人名出现的上下文进行某种划分，使得在同一分区里的人名，在真实世界里“指称”同一个人。在文本中，对人物的区分作用较大的特征包括人名、人物的属性信息以及人名周边的普通词场等。值得说明的是，在人名周边的语句中，与该人经常共现的其他人名，即该人物的交往人群也是一个重要的区分特征。

人物模型的同一性判断和数据融合是相辅相成的。同一性判断是数据融合的前提和基础，只有进行同一性判断，才能为一个人物的数据融合提供信息来源。数据融合是提高同一性判断性能的关键，因为关于一个人物的初始信息是少量和片

面的，只有不断地通过数据融合来得到充实，才能取得代表该人物的关键性特征，从而为同一性判断过程带来重要的影响。因此，本章把人物实体的同一性判断问题和数据融合问题是放在一起来研究的。

11.2　人物模型的特征表示

人物模型主要由四部分组成，即 Person＝{N，P，Q，R}，其中 N 为人物名称（姓名、别名、称谓等）集合，$N=\{N_1,N_2,\cdots,N_n\}$；P 为人物基本属性集合，$P=\{P_1,P_2,\cdots,P_k\}$；Q 为人物介绍性信息集合，$Q=\{Q_1,Q_2,\cdots,Q_m\}$；R 为词场集合，$R=\{R_1,R_2,\cdots,R_l\}$。

设 $X=\{X_1,X_2,\cdots,X_N\}$ 为数据预处理阶段得到的人物模型片段集合，对于 $X_i(1\leqslant i\leqslant N)$，就关注的姓名和人物属性（性别、民族、出生年月、籍贯、家庭住址、职务、家庭关系）以及词场而言，姓名$\in N$；{民族、出生年月、籍贯}$\in P$；{家庭住址、职务、家庭关系}$\in Q$；{人名所在的句子、段落、文本，跟该人名经常共现的其他人名}$\in R$。

11.2.1　属性特征的表示

在集合 N、P、Q 和 R 中，N、P 和 Q 为属性特征集，然而每个属性特征的维度是不同的。设 DOM 为属性特征维度，则 $\mathrm{DOM}(N_i)=1(1\leqslant i\leqslant n)$，因为人物的称谓往往是唯一的；$\mathrm{DOM}(P_i)\geqslant 1(1\leqslant i\leqslant k)$，因为人物的基本属性信息，如民族会有多种情况；$\mathrm{DOM}(Q_i)\geqslant 1(1\leqslant i\leqslant m)$，因为人物的每项介绍性信息，如职业同样变化多样。对于同一个人物而言，集合 $P=\{P_1,P_2,\cdots,P_k\}$ 中每个小项值是唯一的，对于采用信息抽取来获取人物模型数据的情况，每一项对应一个单槽值；集合 $Q=\{Q_1,Q_2,\cdots,Q_m\}$ 中每个小项都是一个可变的集合，每一个小项都对应信息抽取中的一个多槽值。

11.2.2　数值特征的表示

在人物模型中，词场集合 $R=\{R_1,R_2,\cdots,R_l\}$ 为数值特征集合。在文本处理中，常用的表示模型有向量空间模型、概率模型和语言模型等。这里采用常见的向量空间模型来表示 $R_i(1\leqslant i\leqslant l)$，将文档视为向量，向量的每一维代表一个信息表达元素，可以是汉语中单个的字、任何语言中的一个词或者短语，也可以是没有实际意义的 n 元组。这些元素通过对数据集合中的每一个文档进行处理得到。不同特征的重要程度不同，用特征 t_k 附加权重 w_k 来进行量化，可表示为$(t_1,w_1;t_2,w_2;\cdots;t_H,w_H)$，简记为 $D(w_1,w_2,\cdots,w_H)$，这时可以说特征 t_k 的权重为 $w_k(1\leqslant k\leqslant H)$。

需要为向量空间模型进行特征选取。首先确定人名周边词场的作用范围。这里考察三种词场的范围，即人名所在的整篇报道(Text)、人名所在的段落(Para)、人名所在的句子(Sentence)。在每种词场中都进行了汉语分词、词性标注和命名实体识别，然后把人物的属性信息词汇排除。在此基础上，进行停用词(性)处理和特征选择。停用词(性)处理主要目的是去除分辨力很低的词语，主要做法是保留词场中的名词 N、动词 V、形容词 J、副词 R 以及命名实体 NE。针对这些保留的词性，再定制一份停用词表，把个别满足词性要求但分辨力差的的词过滤掉。然后，采用基于文档频率的特征选择方法，把出现频率特别高和特别低的词过滤掉，只保留 DF 值满足 $2 \leqslant DF \leqslant N/3$($N$ 为全部文档个数)的词汇。

对于人物追踪来说，词场中的人名 NR 具有特殊意义，因为它们在一定程度上能够代表该人物与其他人物的密切程度。因此，把词场中的词表示成两个向量，即普通词场向量和共现人名向量。针对每个向量，采用了不同的权重计算方法。

1. 普通词场向量的权重计算

普通词场权重计算采用 TFIDF 计算方法，权重估算公式为

$$w(t,\vec{d}) = \log(\mathrm{tf}(t,\vec{d})+1) \times \log(N/n_t + 1.0)$$

其中，$w(t,\vec{d})$为特征词语 t 的权重；$\mathrm{tf}(t,\vec{d})$为特征词语 t 在文档 $\vec{d}$ 中出现的频率；N 为文档总数；n_t 为训练文档集合中出现 t 的文档数；$\log(\mathrm{tf}(t,\vec{d})+1)$反映的是特征 t 在文档中的重要程度，不同文档中的同一特征 t 的 $\log(\mathrm{tf}(t,\vec{d})+1)$值不同；$\mathrm{idf}(t) = \log(N/n+1.0)$反映的是特征 t 在整个向量空间中相对于其他维的重要程度，每个特征的 idf 值是在样本训练时计算出来的，只要训练集相同，不同的文档同一个特征的 idf 值相同。

2. 共现人名向量的权重计算

采用向量空间模型来描述一篇文档时，往往把该文档中的词汇作为特征，并为每个词汇赋予一定的权重。类似地，人名的共现标识了不同人物之间关系的密切程度，因为每一个人物都有一定的交往人群。对于两个同名的人物来说，在一定程度上可以利用他们交往人群的不同来区分。规定当两个人名出现在文档中同一个段落中时，构成共现关系。

与词场向量类似，这里引入共现人名向量的概念。共现人名向量描述的是人物间关系的信息，它把与当前人名共现过的其他人名作为特征。影响特征权重的因素有两个，首先是人名频率(name frequency，NF)，表示人名的共现次数越大，该人名与当前人物的关系越倾向于密切；其次是倒排人物频率(inverse person frenquecy，IPF)，表示某一人名与其他人名共现得越多，他/她的交际面就越广，那么他/她与当前人物的关系越倾向于不密切。

共现人名向量中每个维度的权重计算公式为

$$w(\text{name},\vec{P})=\log(\text{nf}(\text{name},\vec{P})+1)\times\log(N'/n_{\text{name}}+1.0)$$

其中，$w(\text{name},\vec{P})$为共现人名 name 的权重；$\text{nf}(\text{name},\vec{P})$为人名 name 与人物 $\vec{P}$ 的名字共现的频率；N'为训练文档集合中的人名总数；n_{name}为训练文档集合中与人名 name 共现过的其他人名数。

11.2.3　各项特征的分布规律

在互联网的真实环境中，人物模型的各项特征往往不是完全的，稀疏的程度也不同，如“性别”、“民族”、“籍贯”、“出生年月”、“家庭关系”、“住址”就比较稀疏。表 11.1是数据预处理阶段提取出来的 4315 个人物模型片段中，各项属性的出现次数和稀疏程度。稀疏程度直接会影响其在人物模型同一性判别中的作用。

表 11.1　4315 个人物模型片段中各个属性的出现次数和稀疏度

	性别	民族	籍贯	出生年月	家庭关系	住址	职务
次数	113	15	77	27	38	25	2301
稀疏度	0.026	0.003	0.018	0.006	0.009	0.006	0.533

11.3　人物模型的相似度计算方法

人物模型 $\text{Person}=\{N,P,Q,R\}$包含多个维度，每个维度的相似度计算方法不尽相同，且它们对人物模型总的相似度计算所起的作用也不同。这里规定在人物模型的同一性判别中，进行相似度计算的两个人物模型的名称集合 N 是相同的。在此基础上，讨论其他维度数据的相似度计算方法。

11.3.1　基本属性的相似度计算

基本属性信息对于一个人物来说是不变的，所以在人物基本属性集合 $P=\{P_1,P_2,\cdots,P_k\}$中，每一项的值都是唯一的，即 $P_j=x_j(x_j\in\text{DOM}(P_j),1\leqslant j\leqslant k)$。为了简化和抽象，把“生日”、“籍贯”等信息进行规格化处理，如“生日”最多保留到月份，“籍贯”只保留省份。基本属性信息，如“性别”、“民族”、“籍贯”、“生日”的相似度可表示为

$$\delta(x_j,y_j)=\begin{cases}1, & x_j=y_j\\ 0, & x_j\neq y_j\end{cases}$$

$$x_j\in\text{DOM}(P_j)\wedge y_j\in\text{DOM}(P_j),\quad 1\leqslant j\leqslant k$$

11.3.2　介绍性属性的相似度计算

一般来说，一个人物的介绍性属性信息 $Q=\{Q_1,Q_2,\cdots,Q_m\}$是可能会发生变

化的，如“住址”、“职位”等，每一项的取值都对应着一个集合。设两个人物模型中的介绍性信息的某一维 Q_j 的取值集合分别为 $X_j=\{x_{j1},x_{j2},\cdots,x_{js}\}$，$Y_j=\{y_{j1},y_{j2},\cdots,y_{jl}\}$，则这两个维度的相似度计算方法为

$$\mathrm{Sim}_Q(X_j,Y_j)=\sum_{i=1}^{s}\delta(x_{ji},Y_j)=\sum_{i=1}^{s}\max\{\delta'(x_{ji},y_{jk})\mid x_{ji}\in X_j,y_{jk}\in Y_j\}$$

相对于人物的基本属性信息来说，介绍性信息的描述形式灵活复杂。例如，对“职位”的描述就可能有“所长”、“中科院计算所所长”和“中国科学院计算技术研究所所长”等多种形式。这种局面是由于汉语名词短语结构的复杂性造成的。汉语语言学界对名词短语结构尚未形成成熟而统一的意见，但汉语的名词短语一般是由中心词和修饰词构成，中心词前边可以有多个限定性定语。从意义上说，限定性定语是对中心词的限制和认定，是在中心词所指的人或事物中限定一个小类。从集合论的观点看，中心词表示一个集合，加上一个限定性定语，就得到这个集合的一个真子集。从词语的语义关系来看，中心词表示一个上位概念，加上一个限定性定语，就确立了一个下位概念，限定性定语越多，整个短语表示概念的位置就越低。这里不对名词短语的结构做深入的研究，只是将名词短语 x 看作是由中心词 $H(x)$ 和修饰词 $A(x)$ 构成，即 $x=A(x)+H(x)$。当 $A(x)$ 为空值时，把 x 看做是短名词短语，相反则为长名词短语。在此基础上，按如下方法来计算两个名词短语的相似度。

① 两个短名词短语，只要中心词匹配，它们的相似度就是 1；否则是 0。

② 两个长名词短语，在中心词匹配的情况下还要计算修饰词的匹配程度。这里以字为单位计算二者修饰词的交集，再除以字的并集，如果比值大于 η，则为 1；否则是计算出的 η 值（η 为预设参数）。

③ 短名词职位和长名词短语，只要它们的中心词匹配，相似度就是 1；否则是 0。

综上，两个介绍性属性 x_{ji} 和 y_{jk} 的相似度计算公式可表示为

$$\delta'(x_{ji},y_{jk})=\begin{cases}1, & H(x_{ji})=H(y_{jk});x_{ji}、y_{jk}\text{ 都是长短语时，}\\ & \mathrm{Overlap}(A(x_{ji}),A(y_{jk}))\geqslant\eta\\ \mathrm{Overlap}(A(x_{ji}),A(y_{jk})), & H(x_{ji})=H(y_{jk});x_{ji}、y_{jk}\text{ 都是长短语时，}\\ & \mathrm{Overlap}(A(x_{ji}),A(y_{jk}))<\eta\\ 0, & H(x_{ji})\neq H(y_{jk})\end{cases}$$

其中，$\mathrm{Overlap}(u,v)=\dfrac{|u\wedge v|}{|u\vee v|}$；$\eta$ 为经验阈值（取 0.32 效果较好）。

实际中，对修饰词中一些出现频率特别高的字（如职位短语中的“市省有限责任公司区县乡”），进行了权重减半，因为出现得太频繁的字对短语的区分力较弱。

“住址”信息的表述方式过于灵活，很难客观地衡量它们的相似度，所以“住址”

信息没有参与人物模型同一性判别的计算。

11.3.3　词场的相似度计算

词场集合 $R=\{R_1,R_2,\cdots,R_l\}$ 中，采用向量空间模型来表示 $R_i(1\leqslant i\leqslant l)$。在向量空间模型中，计算两个向量之间的相似度有多种方法。最简单的是直接计算两个向量的内积，最常用的是计算向量的夹角余弦函数。除此之外，还可以采用距离计算公式，如明考斯基距离、欧式距离、曼哈顿距离、Mahalanois 距离和 Lance 距离等。我们采用最常用的夹角余弦的计算方法，即

$$\mathrm{Sim}(X,Y)=\mathrm{Cos}(X,Y)=\frac{\sum_i x_i * y_i}{\sqrt{(\sum_i x_i^2)*(\sum_i y_i^2)}}$$

11.3.4　人物模型相似度计算

人物模型 $\mathrm{Person}=\{N,P,Q,R\}$ 包含多个维度，因此计算两个人物模型的相似度会有很多的方法。最简单的就是不考虑其他维度，只根据词场集合 R，采用夹角余弦来计算。然而，仅根据词场的特征就把人物模型的同一性判定做好，是不易的。因为同名的两个人物，描述他们的文本特征往往非常相似。同一个人物可能存在着方方面面的报道，可能涉及不同领域的内容，词场模型的特征很难把同一个人物的信息有效联系在一起。因而，词场集合本身无法有效解决同名判别的问题。这里试图综合利用人物模型多维度的特征来克服词场集合的缺陷。在人物模型 $\mathrm{Person}=\{N,P,Q,R\}$ 中，属性集合 P 和 Q 中的信息对人物的标识作用是很强的，所以可以通过信息抽取结果之间的匹配产生一些紧密的团，以这些紧密团为“枢纽”，再通过词场相似度的计算来产生最终的结果。同名人物判别方法如图 11.1 所示。

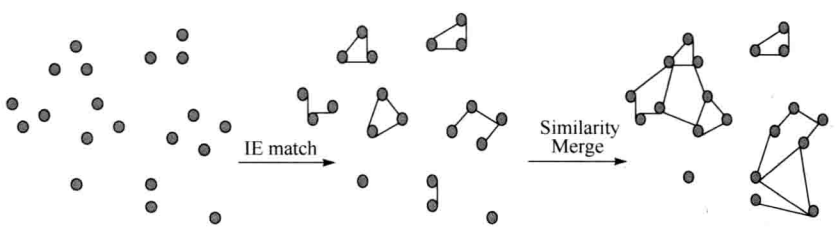


图 11.1　同名人物判别方法示意图

从本章所关注的几个人物属性来看(表 11.1)，“性别”、“民族”、“籍贯”、“出生年月”、“亲属关系”是非常稀疏的(稀疏度在 0.03 以下)，职位的稀疏度是 0.533。属性特征对人物的标识力很强，可以形成如下强匹配的规则。

规则 1：如果姓名相同的两个人物模型，生日相匹配（精确到月份），或亲属关系匹配其一，或长职位匹配其一，则这两个人物模型指向同一个人。

规则 2：如果姓名相同的两个人物模型，性别、民族、籍贯、出生年有其一相抵触，则这两个人物模型指向不同的人。

人物模型中不稀疏的项包括“职位”、“共现人名”、“普通词场”等。对于两个同名的人物模型 X 和 Y，先利用规则 1 和规则 2 进行判断，如果两条规则都不满足，则分别计算相似度 $\mathrm{Sim}_{\mathrm{Position}}(X,Y)$、$\cos_{\mathrm{word}}(X,Y)$、$\cos_{\mathrm{name}}(X,Y)$，然后考虑如何把三者融合起来。假定“职位”、“共现人名”、“普通词场”三个因素是独立的，采用斯坦福确定性理论（Luger，1997）进行三者融合。斯坦福确定性理论定义了一种衡量确信度的方法，能够用一些简单的规则把几个独立的因素融合在一起。假定 E_1、E_2 和 E_3 是事务 B 的三个独立的因素，CF 代表确信度，则事务 B 的确信度 $\mathrm{CF}(B)=\mathrm{CF}(E_1)+\mathrm{CF}(E_2)+\mathrm{CF}(E_3)-\mathrm{CF}(E_1)\times\mathrm{CF}(E_2)-\mathrm{CF}(E_1)\times\mathrm{CF}(E_3)-\mathrm{CF}(E_2)\times\mathrm{CF}(E_3)+\mathrm{CF}(E_1)\times\mathrm{CF}(E_2)\times\mathrm{CF}(E_3)$。例如，如果三个因素对判断一件事务的可信程度分别是 88%、74% 和 66%，则它们混合起来后的可信度为 $88\%+74\%+66\%-88\%\times74\%-88\%\times66\%-76\%\times66\%+88\%\times74\%\times66\%=98.93\%$。事先训练三项各自对应的阈值 $\mu_i(0<i<4)$，认为 E_i 的相似度只要达到该值，两个模型指向同一个人物的可信度就是 100%，即

$$\mathrm{CF}(E_i)=\frac{\mathrm{Sim}_{E_i}}{\mu_i},\quad \mathrm{CF}(E_i)\in[0,1]$$

训练 μ_i 的方法是按每项因素对训练集合单独聚类，在保持很高的准确率（98%以上）的前提下，选取使得召回率尽量高的那个阈值。经过训练，“职位”的阈值取 3；“共现人名”的阈值取 0.5；“普通词场”的阈值取 0.25。

11.4 人物模型的同一性判别与合并

同一性判别的算法分为两步。

① 利用前述的规则 1 和规则 2 把同名的人物模型片段聚成若干紧密的团（这里称之为微类）。

② 利用凝聚式层次聚类的方法对这些微类进一步聚类，主要步骤如下。

Step1，将所有的微类各自单独形成一个簇。

Step2，从现有所有的簇中选择最近（或者最相似）的两个簇，进行合并。

Step3，如果只剩下一个簇或者达到终止条件（所有簇的相似度都低于阈值），聚类结束；否则，返回 Step2。

在人物同一性判别同时，还要把代表相同人物的模型片段进行合并。其中，词场向量的合并比较简单（可采用中心法表示，直接计算特征权重的和平均即可）。

对属性信息的合并，涉及信息抽取中结果模板合并的问题，信息抽取结果模板的合并多采用基于规则的方法，即先确定模板槽类型，再对每一个类型比较两个候选者，看是否是共指、连续或包容关系。采用基本属性的相似度计算公式和属性的相似度计算公式进行属性去重，即如果两个属性的相似度为 1，则它们就是重复的，将其排除掉；否则，保留下来。合并过程结束后，要进行聚类结果的呈现，把同名人物的团并列排放在一起，这样便于比较和浏览。对于一个具体的人物，我们制定了人物简历模板，把识别出的各项信息填充进模板里，从而形成该人物的简历。模板的内容如图 11.2 所示。

<姓名>，<性别>，<民族>，<出生年月>生，<籍贯>人，家住<家庭住址>。
主要亲属有{<亲属姓名>}。
现任<职务>。曾任{<职务>}。
相关人物{<共现人名>}。

图 11.2　人物简历模板结构

11.5　实验结果与分析

11.5.1　数据集与评测方法

这里考察算法对汉语新闻网页中人物同一性判别的效果，数据集的收集仍采用元搜索的方式，从百度新闻搜索引擎上采集查询的返回结果，查询的输入信息是 20 个人名，所选人名力求在所属行业、知名程度、同名程度上达到均衡。原始网页经过进一步的噪声过滤（把新闻正文中不含输入人名的网页都过滤掉）和网页去重，剩下 5874 篇。标注好同名人物的答案，把语料平均分成 5 份，分别拿其中 4 份做训练，1 份做测试，采用 5 次交叉验证的方式，最后的结果取 5 次结果的平均。

采用基于人工标注类的评测方法，对于数据集中某个人工标注的主题 T，假设在自动聚类形成的类层次结构中存在一个与之对应的簇 C。为了发现 C，需要遍历聚类谱系图，计算准确率、召回率和 F 值，以 F 值最大的簇作为与该主题 T 对应的簇。对于任何人工主题 T 和聚类簇 X，计算公式为

$$\text{Precision}(X,T)=\frac{|X\cap T|}{|X|}$$

$$\text{Recall}(X,T)=\frac{|X\cap T|}{|T|}$$

$$F(X,T)=\frac{2P(X,T)*R(X,T)}{P(X,T)+R(X,T)}$$

对于每个人工主题 T，有

$$F(T)=\max_{X\in \text{CluResult}}\{F(X,T)\}$$

其中，CluResult 表示所有聚类结果簇的集合。

最终 F 值为

$$F=\frac{\sum_{T\in \text{TopicHand}}(|T|*F(T))}{\sum_{T\in \text{TopicHand}}|T|}$$

其中，TopicHand 表示所有人工标注主题。

11.5.2 实验结果

1. 人物模型各维度比较结果

这里采用聚类的方法进行人物模型的同一性判别。在普通的文本聚类中，常见的方法是将文本转化成向量空间模型后进行聚类。本章提出的人物模型中，普通词场的维度与此有类似，但这里需要确定人物对应的普通词场的最佳范围。这里着重考查四种人物对应的普通词场的范围，即全篇报道(Text)、人物对应语段(Para_ext)、人名所在段落(Para)、人名所在句子(Sentence)。其中，人物对应语段是指按照本书 10.2.5 节“人物对应语段的确定”介绍的方法来确定的人物在文本中的作用范围。

我们进行了两组实验。实验一是单独采用普通词场，考察四种词场范围对人物同一性判别的效果；实验二是先按照本章前述的规则 1 和规则 2，把语料形成若干紧密的团(这个过程标识为 IE)，再用普通词场进行判别，考察四种词场范围的效果。相关实验的结果如图 11.3 所示。通过图 11.3 的结果可以发现。

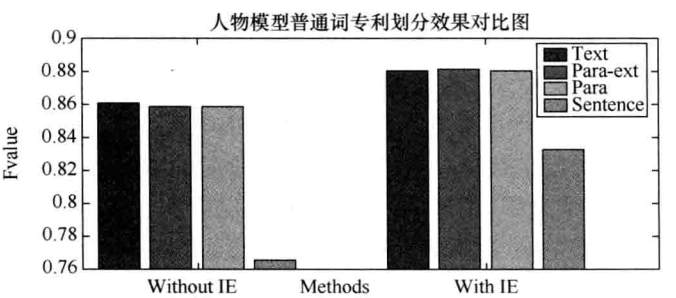


图 11.3 人物模型的普通词场划分效果对比图

① 采用 IE 匹配的方法后，不管是采用何种范围的词场，效果都有显著的提高。

② Para_ext 似乎是人物模型普通词场范围的最佳选择。这是因为 Para_ext 与 Text 相比，在引入 IE 匹配规则前后，效果各有胜负，但是 Para_ext 的计算代价

明显比 Text 低。Para_ext 的效果始终略胜于 Para，尽管 Sentence 的计算代价最小，但它的效果比较差，可能是 Sentence 本身所含的信息量太小，无法对人物进行有效标识。

③ 尽管 Para_ext、Text 和 Para 的效果有所差别，但相差并不明显。这说明单独调整普通词场这一维度对人物同一性判别效果的提高是有限的。

因此，在 Para_ext 基础上，逐渐引进人物模型其他维度的特征，比较了四个系统。

第一，Para_ext，单独采用普通词场向量。

第二，Para_ext＋Position，普通词场向量和职位属性采用“斯坦福确定性理论”进行混合。

第三，Para_ext＋Position＋Fellowship，简称为 Combine，即普通词场向量、职位属性和共现人名向量采用“斯坦福确定性理论”进行混合。

第四，IE＋Para_ext＋Position＋Fellowship，简称为 All，即先采用 IE 匹配规则形成紧密的团，再采用 Combine 模型。

表 11.2 和图 11.4 展示了这四个系统的效果比较。

表 11.2　人物模型各维度效果对比统计表

Run	Precision	Recall	*F*-Measure
Para_ext	0.8841	0.9048	0.8586
Para_ext＋Position	0.9066	0.9081	0.8790
Combine (Para_ext＋Position ＋Fellowship)	0.9309	0.9042	0.8901
All (IE ＋ Para_ext＋Position ＋Fellowship)	0.9512	0.9243	0.9124

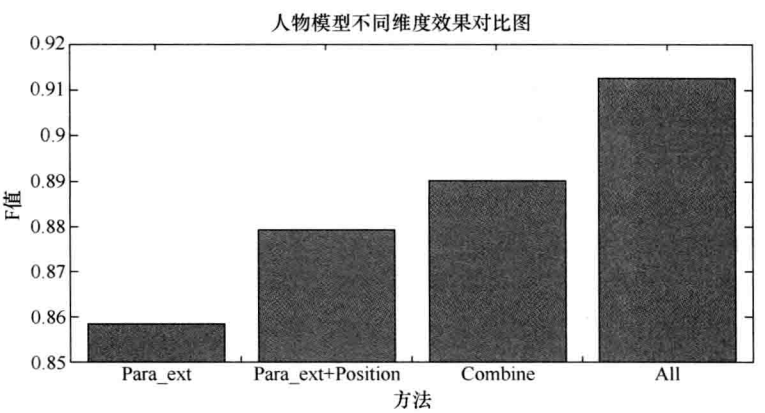


图 11.4　人物模型不同维度效果对比图

由此可见，人物各维度的混合模型能够有效提高人物同一性判别的效果，因为描述一个人物的报道往往包括多方面的信息，只有通过建立人物模型，有效地利用人物的各种类型信息，才能综合体现出人物的本质特征，进而有效地对人物进行判别。

2. 相关算法比较结果

Wan 等(2005)中提出的 WebHawk 是一个对互联网真实环境下的数据进行人物消歧的系统，它同样采用层次凝聚式聚类的方法，主要利用了三方面的特征，即词汇特征、命名实体特征和属性特征。词汇特征主要是文本中的普通词汇。命名实体特征是通过词法分析软件识别出的命名实体，不管它是不是对应于当前的人物。属性特征是通过信息抽取的方法识别出的人物属性信息。WebHawk 对这三类特征分别计算相似度，然后运用一种感知机方法(PAUM)将三者融合起来。WebHawk 在英文网页的人物判别实验中取得了不错的效果，并通过用户的使用，验证了其性能超过了目前搜索结果聚类引擎 vivisimo。我们把 WebHawk 的同名判别方法应用在汉语语料上，来与我们提出的 Person Model 方法进行对比。为了比较两种算法的关键因素，我们把 WebHawk 的多因素融合步骤采用斯坦福确定性理论来进行，修正后的算法称之为 WebHawk_like。我们在百度搜索引擎返回的汉语网页结果中进行了对比实验，结果如表 11.3 所示。

表 11.3 汉语语料人物同一性判别中不同系统的效果对比

System	Precision	Recall	F-Measure
Baidu	0.2313	1.0000	0.2464
WebHawk_like	0.9201	0.9172	0.8904
Person Model	0.9512	0.9243	0.9124

从表 11.3 可以看出，通用的搜索引擎不区分同名人物的问题，即针对一个人名的查询，把所有的返回结果都认为属于同一个人物，这样的准确率只能达到 0.2313。WebHawk_like 和 Person Model 都能较好解决同名人物判别的问题，Person Model 效果略好，主要原因在于 Person Model 能够更科学地反映人物的本质特征，能够把人物多方面的信息妥善地进行融合，并充分考虑到了人物特征的分布规律。图 11.5 和图 11.6 分别是对百度搜索引擎的搜索结果，采用 Person Model 方法处理前后，典型的 10 个人名的同一性判别效果。

3. 实验结果说明

在人物同一性判别实验中，使采用最为简单的基于整个文档的向量空间模型

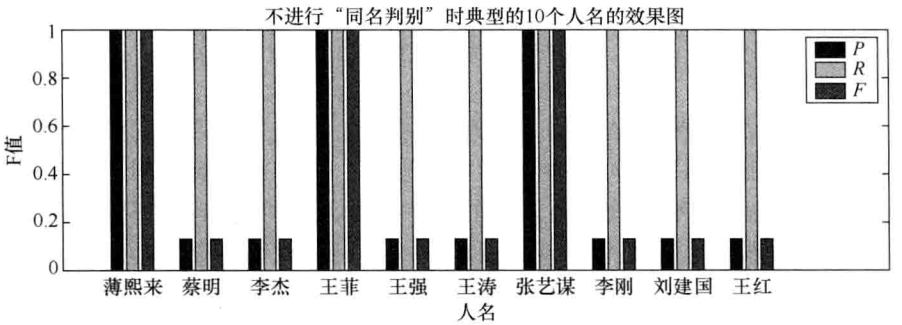


图 11.5　不进行“同名判别”时典型的 10 个人名的效果图

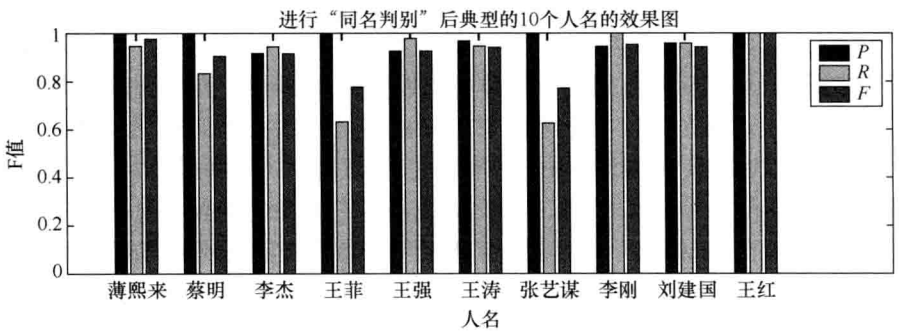


图 11.6　进行“同名判别”后典型的 10 个人名的效果图

聚类方法，也能取得比较好的效果（$P=0.8863$，$R=0.9041$，$F=0.8592$），而采用复杂的多维度混合方法后，性能只提高了 6%。我们认为这与本实验建立的语料有关，原因主要有以下几点。

① 新闻网页中出现的人物报道并不能体现人物的方方面面的生活状态。例如，“江泽民”可能只会报道他的政治活动等，对其生活方面的描述可能很少，这就使得描述一个人的报道在内容上相似度比较高。

② 新闻网页并不是对社会上每个人物都报道，被报道的人物行业特征也比较明显，这就使得同名人物的报道内容相似度并不高。

③ 建立的语料收集的是百度新闻搜索引擎的返回结果，这些结果的时间跨度比较短。在这么短的时间内，人物的状态往往没有大的变化，对其报道的文字特征当然也就没有太大的变化。

如果把提出的人物模型混合计算的方法应用在描述更加复杂的语料中，跟踪更长一段时期内人物的报道，将能够发挥更大的作用。

11.6 基于宏观粒度的事件组织

经过数据预处理和人物的同一性判别，能够把描述一个人物的信息有效地归结在一起。然而描述一个人物的信息往往是杂乱无章的，需要对同一人物的信息进行整理，梳理出脉络，实现对该人物的追踪。对人物的追踪，归根到底还是跟踪该人物参与的事件信息，具体来说有两种方式：一种是“编年史”的方式，即以时间为主线制作人物的“大事年表”；一种是“叙事”的方式，即以事件为主线制作人物的“传记”。从目前事件信息组织的技术水平来看，一种方法可以识别出具体的事件角色框架，但方法本身往往局限于特定的领域，称为微观粒度的事件组织；另一种方法以文本或段落为粒度单位，可以对非限制领域事件信息进行大规模处理，但分析出的事件特征往往比较模糊，称为宏观粒度的事件组织。本节介绍宏观粒度的事件组织技术（话题识别与跟踪）、事件的属性特征对话题识别的影响、话题的层次化组织方法，利用话题的层次化组织来挖掘话题之间的关系。宏观粒度的事件组织是以文本或段落作为描述事件的基本单位，借用信息检索的技术把具有事件内在联系的报道组织在一起。尽管这种利用宏观特征的处理方法不如细粒度事件挖掘那样精细和深入，却可以克服事件信息抽取方法的限定领域的弊端，对各种类型的事件进行大规模地高效处理。

11.6.1 宏观粒度事件的特征

为了对事件信息进行有效处理，有必要了解清楚事件的特征。从事件自身的发展、事件之间的关系和事件的演变来概括事件的属性，主要有以下三方面特征。

（1）事件是有生命周期的

事件的生命周期有长有短，在生命周期内的各个阶段往往会呈现不同的分布规律。按照出现规律，事件可分为突发性事件和持久性事件。

（2）事件是有层次的

事件可由子事件构成，子事件构成的方式分为同构模式和非同构模式两种。某个具体事件可以用三个维度来刻画，即时间维，空间维，语义维。从任一个维度来看，事件及其子事件之间都能产生包含关系。

① 从时间维来看，发生时间范围广的事件包含发生在其间的子事件。

② 从空间维来看，发生空间范围广的事件包含发生在其间的子事件。

③ 从语义维来看，语义层次高的事件包含语义层次低的事件。

实际上，具体事件及其子事件之间的包含关系往往是同时从这三个维度出发来考察的。我们说一个事件包含一个或多个子事件，是指该事件对其子事件来讲发生的时间范围宽、空间范围广、语义层次高。

(3) 事件是可变的

事件未必如预想的那样达成或完结，而是往往出现变局。预想的达成或完结，可称之为流变。此外还可能出现两种情况，一种是“裂变”，即一个事件分解成多个独立的事件分别展开，一种是“蠕变”，即事件逐步“变味儿”。事件发生时，往往会出现多方面的报道。事件下面还包含着子事件。由于事件的知名程度不同和包含的子事件数目不同，对它的报道数量会有所不同。对于同一个事件的不同发展阶段，各大新闻媒体的报道数量也不尽相同。

11.6.2　针对事件特点的话题识别方法

话题识别处理的对象是热点话题的新闻报道，所用方法是否符合事件的特点，对系统性能的提高极为重要。对话题很好地识别，需要有效的事件组织技术，这是话题识别的基础。这里主要根据事件的特点来挖掘一些方法，着重关注话题的时序属性、事件角色特征词对话题识别的影响、自然语言处理技术在话题识别中的应用。

在自由文本处理领域，把两个属于同一类型的不同事件区分开是很难的，如两次不同的火车事故。描述这两个事件的词汇如此相似，以致通常的文本计算策略很难把它们区分开；同一事件内部的报道相似度往往并不高，因为事件有不同的发展阶段，且往往会出现“变局”。如何把事件的起承转合关系有效地识别并组织起来，是文本处理领域的一个挑战。由于事件信息处理有这样的特殊性和难点，所以话题识别不能完全遵循通常的文本聚类方法，必须针对事件的特点进行策略调整和方法创新。下面介绍我们提出的核心方法和策略。

1. 基于分组策略的识别方法

在话题识别与跟踪中，时序是一个重要特征。随着事件的逐步展开，对事件的报道也是按照时间先后顺序逐步出现的。一般来说，讨论同一事件的报道在时间上往往是相近的，时间间隔越远越倾向于描述不同的事件。按照出现的模式，事件可分为突发性事件和持久性事件。突发性事件的特点是持续的时间较短，但在出现的时期内报道的数量却很多，如微博热点话题。持久性事件的特点是持续时间长，但报道的数量在各个时间段内可能分布不均匀，在有的时间段内可能呈现突发性的趋势。

在传统的话题识别任务定义中，报道要求按照时序进行处理，并且允许有一定的时间延迟。我们考察了时序信息在话题识别与跟踪系统中的作用，对文档按照固定间隔进行分组，组内优先聚类，考察了按照时序分组和不按时序分组，以及不同分组间隔的大小对系统的影响。在对比实验中发现对文档按照报道的时间顺序进行分组，组内优先聚类，再把聚类结果合并的策略，能提高系统的效果。方法步骤如下。

① 提取报道的时间信息，把报道按照时间顺序排序。

② 按照一定的时间间隔将报道分成若干组。

③ 在每一组内，采用自底向上的层次凝聚式聚类方法进行聚类，利用阈值控制生成的“微类”数量。

④ 去掉各组之间的间隔，但仍然保持各“微类”结果的时间顺序，采用单遍聚类算法将各“微类”结果进行合并。

2. 基于概念空间的命名实体权重计算方法

要有效识别出属于同一类型却不同事件的报道或属于同一事件却不同类型的报道，必须寻找出真正具有代表性的特征。这里把文本中的词汇划分为两种类型，即命名实体(NE)和其他普通词(N)。命名实体主要包括人名、地名、机构名等。普通词主要包括名词、形容词和一些特殊的符号。从直觉上来看，当报道两个不同的火车事故时，事故所发生的时间、地点以及涉及的人物会有比较大的区分能力。我们认为对命名实体只进行加权是不够的，解决相关词之间的不匹配现象也极为重要。在文本中，代表同一命名实体概念 C_i 的词通常不止一个，而往往是一个词的集合 $C_i=\{t_{i1},t_{i2},\cdots,t_{ik}\}$。例如，代表“中科院计算所”这个概念的命名实体词可能有“计算所”、“中科院计算所”、“计算技术研究所”、“中国科学院计算技术研究所”、“ICT”等，这种现象是常见的。由于命名实体识别技术的限制，代表同一命名实体概念的词 t_{ij} ($1\leqslant j\leqslant k$)，可能被标注为命名实体词 $\mathrm{NE}_{t_{ij}}$，也可能被标注为普通词 $N_{t_{ij}}$。如果忽视这些现象，就会造成概念空间的不匹配，命名实体的加权策略又会使概念失配的影响加剧。为此，提出基于概念空间的命名实体权重计算方法，主要是把代表同一概念的词汇统一起来计算权重，即

$$\mathrm{IDF}_{C_i}=\frac{N}{\sum_{j=1}^{k}n_{t_{ij}}},\quad n_{t_{ij}}\text{ 为出现 }t_{ij}\text{ 的文档数}$$

$$\mathrm{TF}_{C_i}=\sum_{j=1}^{k}(\mathrm{TF}_{\mathrm{NE}t_{ij}}+\mathrm{TF}_{Nt_{ij}})$$

$$W_{C_i}=\alpha\log(\mathrm{TF}_{C_i}+1)\cdot\log(\mathrm{IDF}_{C_i}+1),\quad \alpha>1$$

其中，W 表示命名实体概念的权重。

利用半自动的方式对代表相同概念的命名实体建立同义库，库中主要收录国家、地区、城市、团体机构等的别称和简称；建立命名实体停用后缀表，出现在表中的命名实体后缀被过滤掉，利用这些资源把命名实体在概念空间上统一起来，再对其进行加权。在 TDT4 中文语料 27 142 篇文档中，命名实体词出现477 277 次，利用概念空间统一起来的有 45 691 个，占命名实体词的 10%；被还原为命名实体的普通词有 33 410 个，占命名实体词的 7%。

3. 指代消解在话题识别中的应用

到目前为止，话题识别与跟踪领域的大部分研究主要还是借鉴信息检索的方法，通过调整某些参数来使这些方法更适合处理话题。但是话题识别与跟踪研究的特殊性，如面向话题，基于时间等，也决定了利用信息检索方法对进一步提升系统的性能是有限的，要想突破必须借助于更多的自然语言理解技术。

我们尝试在话题识别与跟踪中引入指代消解技术。指代消解是确定指代词与先行词之间的关系，从而明确指代词指代什么对象。在新闻报道的叙述中，主要的事件角色特征词往往只出现一两次，更多的是利用指代词对该特征进行引用。考虑一篇新闻报道的事件角色特征集合 $T=\{T_{C1},T_{C2},\cdots,T_{Cn}\}$，其中 $T_{Ci}=\{P_i,n_{i1},n_{i2},\cdots,n_{ik}\}(1\leqslant i\leqslant n)$，$P_i$ 为角色 T_{Ci} 的先行词，$n_{ij}(1\leqslant j\leqslant k)$ 为 P_i 的指代词。在以往的系统中，指代词要么与先行词完全独立进行处理，要么作为停用词性被过滤掉，即 $\mathrm{TF}_{T_{Ci}}=\mathrm{TF}_{P_i}$（TF 为词频，$1\leqslant i\leqslant n$），这就削弱了相关角色在系统中的地位，造成事件角色识别的不均衡。为此，这里采用基于指代消解的事件角色词频计算方法，主要是把指示代词引入权重计算中，即

$$\mathrm{TF}_{T_{Ci}}=TF_{P_i}+\beta\sum_{j=1}^{k}\mathrm{Dis}(P_i,n_{ij}),\quad 0<\beta<1$$

$$\mathrm{Dis}(P_i,n_{ij})=\begin{cases}\mathrm{Count}_{\mathrm{sentence}}(P_i,n_{ij}), & n_{ij}\text{ 是代词}\beta\cdot\mathrm{Count}_{\mathrm{sentence}}(P_i,n_{ij})<1\\ \dfrac{1}{\beta}, & \text{其他}\end{cases}$$

其中，$\mathrm{Count}_{\mathrm{sentence}}(P_i,n_{ij})$ 为先行词 P_i 与指代词 $n_{ij}(1\leqslant i\leqslant n,1\leqslant j\leqslant k)$ 之间所夹的句子个数；β 为预设参数（取 0.3）。

在这种方法中，指示代词与先行词的距离越大，则表明它越能独立的代表一定的意思，因而权重越高。

我们主要针对人物姓氏和第三人称代词进行了指代消解的处理，所用的指代消解系统采用基于规则的方法，综合应用了性别、单复数和距离等属性信息。在 TDT4 汉语语料的 27 142 篇报道中，人物全名出现了 142 601 次，姓氏或名字单独出现的情况有 32 282 次，第三人称代词出现了 56 653 次，我们对这些词采用了基于指代消解的词频计算方法。

11.6.3　基于多层聚类的话题层次化组织方法

不同的话题往往不是孤立的，它们之间存在一定程度的关系，话题的粒度也是一个很难判断的概念。为了解决这个问题，一个思路是把话题按照层次化的关系组织起来。关于这项研究，在话题识别与跟踪国际评测里有一项标准的评测任务，即层次化话题识别（hierarchical topic detection，HTD）。层次化话题识别在评测

中是一项全新的任务，是传统话题识别任务的扩展。这里介绍对层次化话题识别问题的研究，首先介绍该项任务的优点及主要解决方法，然后介绍我们的思路和方法。

1. 层次化话题识别的优点

层次化话题识别是对原来的话题识别任务的扩展。传统的话题识别任务把所有的话题都折中在同一粒度层次上，这样并不符合话题的特点。事实上，话题是经常处在不同粒度层次上的，例如“亚洲金融危机”包含“印尼股市暴动”、“日本银行破产”等。与以往的话题识别任务相比，层次化话题识别力图克服两个不符合实际情况的假设，即假设所有的话题都处在同一粒度层次上，假设每一个文档至多属于一个话题(NIST, 2004)。

与传统的话题识别任务不同，层次化话题识别系统把文档组织成有向无环图(DAG)而不是扁平式结构，通过这种方式来更好地反映话题的层次包容关系。从应用角度来看，层次化话题识别突破了传统的“一类一团”的结果呈现方式，用户可以通过图形化的层次结构进行浏览，有利于减少阅读工作量。同样，层次化话题识别本质上是一个非监督式机器学习的过程，文档聚类是其中很自然的应用。在层次化话题识别中，应用针对事件的信息处理技术把同一粒度层次的话题识别出来后，还要把处于不同粒度层次的话题节点合理组织，形成科学的层次结构。

下面介绍一下代表话题之间关系的有向无环图的拓扑结构。

2. 有向无环图的拓扑结构

层次化话题识别把话题组织成有向无环图的结构，该图的具体含义如下。

① 每一个图节点都用一个唯一的字符串标识。

② 每一个图节点都包含若干报道或其他图节点。

③ 一个节点所包含的所有报道都自动包含在它所有的祖先节点中。

④ 报道可存在于任何节点中，但不可冗余地直接存在于它的祖先节点中。

⑤ 节点之间的边不可形成环。

⑥ 有且仅有一个根节点，作为其他所有节点的总领。

⑦ 内部节点对应一个置信度数值，指示其对应话题的聚合程度。

有向无环图结构如图 11.7 所示。

3. 层次化话题识别的主要方法

层次化话题识别任务在第七届话题识别与跟踪国际评测中首次出现。作为首次任务的参赛者，我们与其他三个参赛队伍一起开展了对这个问题的研究。另外三个参赛队伍分别是 TNO、UMASS、香港中文大学(CHUK)。每个参赛队伍所

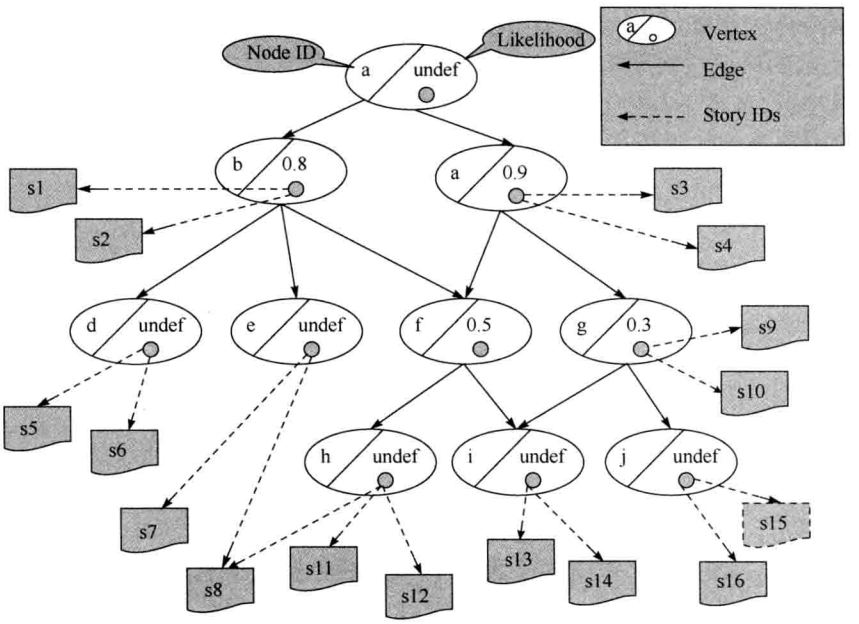


图 11.7　话题层次结构

采用的方法都各有特色。香港中文大学采用分裂式 K-means 算法来产生话题的有向无环图，其中 K-means 算法用来把报道组织成一定粒度层次的话题节点，然后采用分裂式聚类算法把粒度层次高的话题节点分割成粒度低的话题节点。TNO 将层次化话题识别分为四个步骤来进行，首先采样 20 000 个报道来建立相似度矩阵，然后用自底向上的凝聚式聚类算法生成一棵二叉树，接下来对二叉树进行适度的剪枝，然后把剩余的报道分别插入到 10 个最相近的节点中。UMASS 在聚类算法中引入一定的约束条件来减少时间复杂度，即采用带有约束条件的1-NN 算法来形成话题节点，用凝聚式聚类算法建立层次结构，使用的约束条件主要是采用有限的窗口，同一新闻来源的报道优先聚类，然后是同一语言的报道，最后是所有语言的报道。

与其他参赛者相比，我们采用的方法也是自底向上的聚类方式，先形成每一粒度层次的话题节点，然后再把各个层次的节点组织起来，称为基于多层聚类的层次化话题生成(multi layered clustering，MLCS)算法。

从有向无环图的结构特点可以看出，根节点是所有话题的总领，包含若干子节点。每个节点即可以包含文档，也可以指向一个或多个更下层的节点。每个节点代表某一粒度的话题，节点的层次越往下，对应话题的粒度越小。

自底向上的文本聚类似乎是形成有向无环图的天然方法。一般地，可以利用不同大小的相似度阈值生成不同粒度层次的话题节点，相似度阈值越大，生成的话题节点粒度就越小。该阈值代表着对应话题节点的聚合程度。如果把通常的单层

文本聚类算法组织成多层聚类，记录各层次间节点的合并过程，就会得到如图11.8所示的话题层次结构。特别地，如果一个节点与两个或多个上一层节点的相似度很接近，且高于其他的节点，则把这个节点同时合并到这几个上层节点中去。基于多层聚类的层次化话题生成算法的基本思想如图11.8所示。

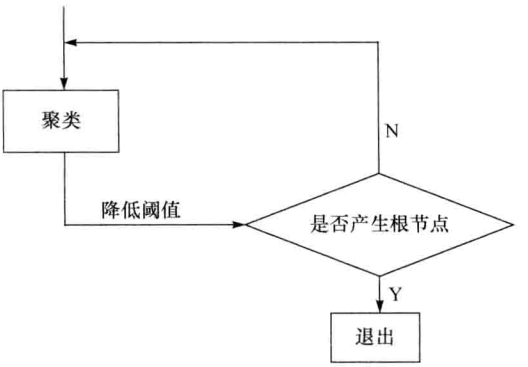


图11.8 基于多层聚类的层次化话题生成算法的基本思想

从多层聚类算法对有向无环图结构的影响来看，聚类的层数决定了有向无环图的深度，每一层聚出的类数决定了有向无环图的宽度。算法采用自底向上的聚类方法，首先从一个较大的阈值开始从最底层对所有文档进行聚类，当该层聚类结束后，阈值减去一定的步长，对聚出的结果再一次进行聚类，依此循环，直至满足终止条件为止。在实际做法中，算法利用阈值的个数控制聚类的层数，利用阈值的大小控制每层的类数。

特别地，基于多层聚类的层次化话题生成算法采用时序分组策略，在最底层对文档按照报道时间间隔进行分组，组内优先聚类。从话题层次化结构的角度来看待这种策略，可以理解为事件是有层次的，事件可由子事件构成；事件可以用三个维度来刻画(时间维、空间维、语义维)，各个维度综合反映了不同大小的事件粒度。利用这种方法，把报道时间相近的文档优先聚合到一起，突出了特征最为明显的时间维的特性。基于多层聚类的层次化话题生成算法描述和示意图如图11.9和图11.10所示(“Agglo.”代表层次凝聚式聚类方法)。

1. 初始化参数θ、q、k、t；令$i=1$。

2. 把报道按照时间顺序排序，再按固定的时间间隔t将报道分成若干组$\{P_1,P_2,\cdots,P_n\}$。

3. 在$P_j(1\leqslant j\leqslant n)$内，采用自底向上的层次凝聚式聚类方法进行聚类，利用相似度阈值θ控制生成的“微类”数量；在“微类”和对应的文档之间生成边。

4. 去掉各组的间隔，但保持各“微类”结果的时间顺序。

5. 如果$i>k$，或者$\theta<0$，转8。

6. 采用单遍简单聚类(Single-Pass)算法将上一层“微类”结果进行合并，利用相似度阈值 θ 控制结果的数量；在新产生的“高层类”和它的子类之间生成边。

7. $\theta -= i \cdot q$，$i++$，转 5。

8. 如果还没产生根结点，则生成根结点和它的边，退出。

图 11.9　基于多层聚类的层次化话题生成算法描述

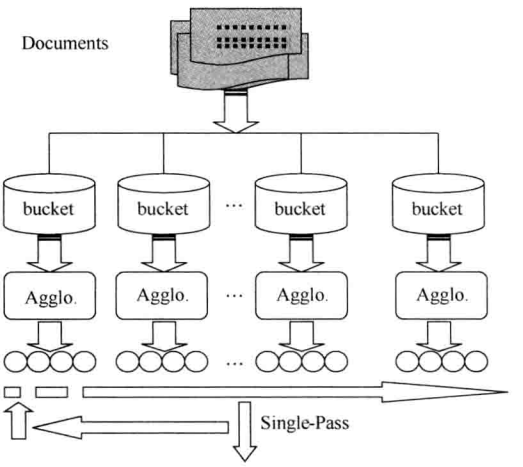


图11.10　基于多层聚类的层次化话题生成算法示意图

在基于多层聚类的层次化话题生成算法中，影响算法的因素主要有相似度阈值 $\theta(0 \leqslant \theta \leqslant 1)$，相似度递减步长 $q(0 < q \leqslant \theta)$，简单聚类最大循环次数 $k(k > 0)$ 和分组时间间隔 $t(t > 0)$。通过调整具体的参数值，可以生成不同结构的有向无环图。

4. 基于多层聚类的层次化话题生成算法的增量模式

基于多层聚类的层次化话题生成算法本身的时间复杂度是输入文档数的立方，在语料规模大时效率较低。为了提高算法的运行效率，我们为算法引入增量模式。首先选取一定规模的文档，并利用算法建立初始的有向图；然后把剩余的文档一批一批地处理，先将每一批文档形成若干微类，对于每一个微类，遍历有向图，把该微类插入到跟它最相似的图节点中去。把基于多层聚类的层次化话题生成算法应用在 TDT2004 的 HTD 任务的跨语言场景中。在评测中，我们选取原始语言是英文的文本为初始文档，把汉语和阿拉伯语翻译成英语的文本作为增量处理的文档，运用增量算法进行处理。

11.6.4　实验结果与分析

下面介绍所采用的数据集和评测指标，然后给出实验结果、评测结果及相应分析。

1. 数据集

TDT4 语料包含时间跨度为四个月的新闻专线和广播语料，包含 98 245 篇报道(英文 28 390 篇，汉语 27 142 篇，阿拉伯语 42 713 篇)。汉语和阿拉伯语语料提供本国语料和翻译语料两种形式，翻译语料采用机器翻译软件翻译成英语。广播语料是通过语音识别软件自动转录而成的。整个语料共有 25 960 个文件，共计约 2.5GB。TDT4 语料主要来源于大型的新闻媒体机构，语料里标注的与话题相关的报道共有 6000 多篇，蕴涵在 100 个话题中。实验中利用 TDT4 的汉语语料。

2. 评测指标

层次化话题识别可以有多种评测手段，其中一些在文献(NIST，2004)里有所介绍，目前主要采取基于最小代价的评测方法。该方法主要考虑识别代价和遍历代价，评测公式为

$$\text{MinimumCost(topic)} = \text{WDET} * [C_{\text{Det}}(\text{topic}, \text{bestVtx})]_{\text{Norm}} + (1 - \text{WDET}) * [C_{\text{Travel}}(\text{topic}, \text{bestVtx})]_{\text{Norm}}$$

其中，bestVtx 是评测系统找到的使得对应话题代价最小的那个节点；WDET 是权重调节参数。

识别代价$(C_{\text{Det}}(\text{topic}, \text{bestVtx}))_{\text{Norm}}$由误报率和漏报率加权求和并归一化而成，其计算公式为

$$(C_{\text{Det}})_{\text{Norm}} = \frac{C_{\text{Miss}} P_{\text{Miss}} P_{\text{target}} + C_{\text{FA}} P_{\text{FA}} P_{\text{non-target}}}{\min\{C_{\text{Miss}} P_{\text{target}}, C_{\text{FA}} P_{\text{non-target}}\}}$$

其中，C_{Miss}和 C_{FA}分别是漏报和误报的代价；P_{Miss}、P_{FA}和 P_{target}是预设参数，用来调节漏报率和误报率在识别代价中所占的比重；$P_{\text{non-target}} = 1 - P_{\text{target}}$。

遍历代价$(C_{\text{Travel}}(\text{topic}, \text{bestVtx}))$是从根节点遍历到 bestVtx 节点所花费的阅读代价，其计算是一个递归累加的过程，即

$$C_{\text{Travel}}(\text{topic}, \text{vertex}) = C_{\text{Travel}}(\text{topic}, \text{parentOf(vertex)}) + \text{CBRANCH} \cdot \text{NumChildren(parentOf(vertex))} + \text{CTITLE}, C_{\text{Travel}}(\text{topic}, \text{root}) = 0$$

在该公式中，根节点的遍历代价为 0；$C_{\text{Travel}}(\text{topic}, \text{parentOf(vertex)})$是当前节点的父节点的遍历代价；NumChildren(parentOf(vertex))是当前节点的兄弟节点的个数；CBRANCH 和 CTITLE 是预设参数，分别表示从兄弟节点中找出最佳节点所花费的阅读代价和阅读当前节点的摘要所花费的代价。最后，利用以下公式对计算得到的$(C_{\text{Travel}}(\text{topic}, \text{bestVtx}))$值进行归一化，即

$$C_{\text{Travel}}(\text{topic}, \text{bestVtx}_{\text{Norm}}) = \frac{C_{\text{Travel}}(\text{topic}, \text{bestVtx})}{(\text{CBRANCH} \cdot \text{MAXVTS} \cdot \text{NSTORIES/AVESPT}) + \text{CTITLE}}$$

归一化因子的含义是每个话题的平均阅读量。MAXVTS 是每篇报道至多归属的话题数；NSTORIES 是语料中全部的报道数；AVESPT 是平均每个话题包含的报道数，各个语料有所不同(在 TDT4 汉语语料的实验中，AVESPT 的实际值为 16；在 TDT2004 评测中，AVESPT 取值 88)。在以上公式中，预设参数取值为 $P_{target}=0.02$、$C_{Miss}=1.0$、$C_{FA}=0.1$、WDET=0.66、CBRANCH=2.0、CTITLE=1.0、MAXVTS=3。

3. 话题识别中不同分组策略的实验结果

基于多层聚类的层次化话题生成算法采用基于时序的分组策略，为了验证时序分组对话题识别代价的影响，这里进行了相应的对比实验。令算法中的参数 $k=1$，$q=0$，令评测公式中的 WDET=1，即不采用层次化的组织结构，也不考虑遍历代价的评测影响因素。考察了不分组、按时序分组与不按时序分组对识别代价的影响，结果如图 11.11 所示。

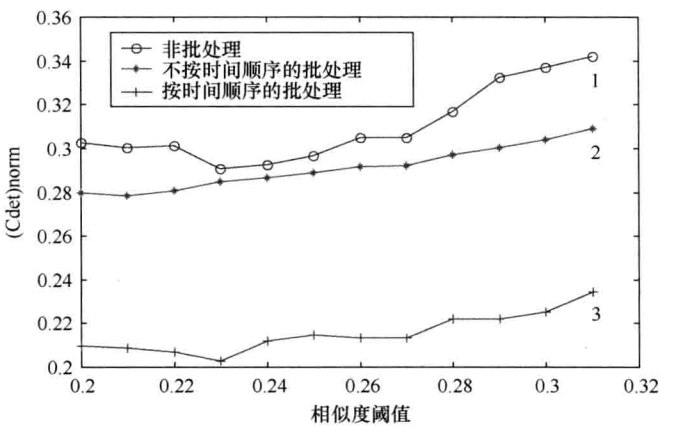


图 11.11　不同分组策略效果对比图

在图 11.11 中，曲线 1 表示不对文档进行分组，一开始就对全部语料进行单遍聚类的结果，横坐标是不同的聚类相似度阈值 θ。曲线 2 是采用时序分组的结果，分组时间间隔为 7 天。在曲线 3 的实验中，仍旧把文档进行分组，组内优先聚类，与曲线 2 不同的是，我们把文档按时间均匀打散，也就是每组内的文档在时间上并不是相邻的，然后按固定的报道数进行分组。结果表明，基于时序分组的策略使识别代价降低了近 30%；当组内文档时间不相邻时，分组策略对效果提高不明显。这里又考察了在时序分组的条件下，不同的分组时间间隔对识别代价的影响，结果分布如图 11.12 所示。

通过实验可以看出，时序分组策略能有效降低系统的识别代价，主要原因是由于它把时间相近的文档优先聚合在一起，提高了属于同一话题的文档的合并机会。

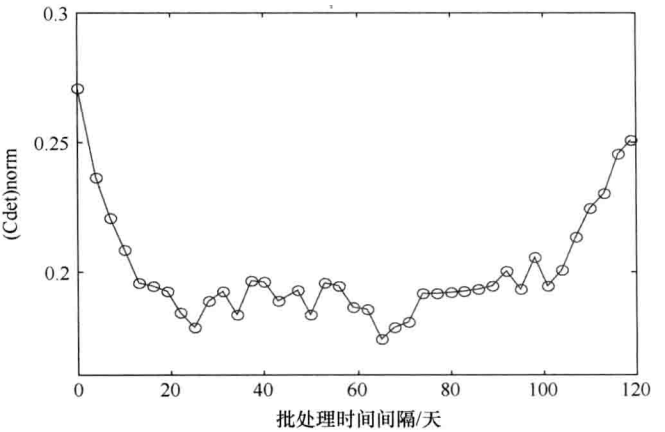


图 11.12 不同分组间隔效果分布图

这说明新闻话题具有比较明显的时序聚合性。

4. 针对事件的话题识别方法的实验结果

为了检验这里提出的话题识别方法对层次化话题识别系统性能的影响，我们构建了原型系统 BASE 和三个改进的系统，即系统 1、系统 2 和系统 3。

BASE 系统采用向量空间模型。在语料预处理阶段，对于中文语料，进行分词、词性标注和命名实体识别；对英文语料，系统进行词性标注和词形还原。特征选择包括两层策略，一是通过常用的禁用词表来过滤禁用词，还建立了禁用词性表，去掉动词、副词、助词等，只保留名词、形容词、特殊符号和命名实体；二是使用文档频率信息，即把 $\mathrm{DF}(t_i) \leqslant 3$ 或 $\mathrm{DF}(t_i) \geqslant N/3$ 的特征词都过滤掉，因为符合这两个条件的特征分布太窄或太广，区分能力不强。权重计算采用常用的 TFIDF 计算公式为

$$W_t(t_i)=\log(\mathrm{TF}(t_i,d)+1)\cdot\log(\mathrm{N}/\mathrm{DF}(t_i,d))+1)$$

其中，词频 $\mathrm{TF}(t_i, d)$ 为特征 t_i 在文档 d 中的出现频度；文档频率 $\mathrm{DF}(t_i,d)$ 为 t_i 在其中出现的文档数；N 为语料总文档数。

相似度计算采用夹角余弦公式。BASE 系统同样利用多层聚类使文档形成有向无环图结构，然而与基于多层聚类的层次化话题生成算法不同的是，没有对文档按照时序分组的策略，而是直接利用单遍简单聚类算法进行多层聚类。

系统 1(MLCS)是在 BASE 系统的基础上，采用提出的 MLCS 算法，即首先对文档按照时序进行分组，组内优先聚类，再用单遍聚类算法进行多层聚类。实验中采用的分组时间间隔 t 为 7 天。

系统 2(MLCS+NE)是在系统 1 的基础上，又采用了基于概念空间的命名实体权重计算方法。

系统 3(MLCS+NE+PR)是在应用系统 2 所有技术的同时，又引进了基于指代消解的特征权重计算策略。

表 11.4 是这四个系统在 TDT4 汉语语料上得到的最好结果(表中 ↑ 标记指示当前结果与 BASE 系统相比上升的幅度)。

表 11.4　针对事件特点的话题识别方法效果对比表

系统	$(\text{Cost(detect)})_{\text{Norm}}$	$(\text{Cost(travel)})_{\text{Norm}}$	MinimumCost
BASE	0.2718	0.2653	0.2696
MLCS	0.1968 (↑27.59%)	0.2139(↑19.37%)	0.2026(↑24.85%)
MLCS+NE	0.1543(↑43.23%)	0.2267(↑14.55%)	0.1789(↑33.64%)
MLCS+NE+PR	0.1487(↑45.29%)	0.2105(↑20.66%)	0.1697(↑37.05%)

可以看出，应用针对事件的处理策略后，系统的效果有了提高。采用了 MLCS 算法的系统 1，对系统效果提升得最为明显，它不仅有效地减少了识别代价，还大大降低了遍历代价；既采用了 MLCS 算法，又采用了基于概念空间的命名实体权重计算策略的系统 2，进一步降低了识别代价和总的代价；在采用系统 1、系统 2 所有技术的基础上，又引进了指代消解技术的系统 3，取得了最好的效果。从识别代价和遍历代价的提升幅度可以看出，采取了有效的事件组织策略后，最主要的是能够把真正属于同一话题的报道组织在一起，从而大大地降低了系统的识别代价，对系统总的代价也能够产生好的影响。

5. 话题层次化结构组织的实验及评测结果

在话题层次化组织算法 MLCS 中，通过调整相似度初始阈值 θ、相似度递减步长 q、简单聚类循环次数 k 可得到不同的有向无环图结构。通过参数 θ、q、k 的不同组合，来判断不同的结构对效果的影响，利用系统 3 在 TDT4 汉语语料上做了相关实验，并用其参加了 TDT2004 层次化话题识别任务的评测。在 TDT4 汉语语料中的实验结果如表 11.5 所示，其中的时间分批间隔 t 取为 7。从实验结果可以看出“平面形”的结构(不分层，即传统的话题识别方法)，当考虑遍历代价时，效果差(因为用户需要浏览的类数太多，以致无法利用有限的精力找到关注的话题)。“梯形”结构效果好一些，但遍历代价还是偏高，同时又影响了识别代价。“锥形”效果最好，即将文档从初始阈值开始逐步分层组织，直到自动生成根节点为止。

表 11.5　MLCS 算法生成的不同话题结构的效果对比表

结构类型	θ	q	k	$(\text{Cost(detect)})_{\text{Norm}}$	$(\text{Cost(travel)})_{\text{Norm}}$	MinimumCost
平面形	0.35	0.00	1	1.0000	0.0000	0.6600
梯形	0.35	0.05	3	0.1959	0.5128	0.2410
锥形	0.35	0.05	$+\infty$	0.1501	0.2130	0.1715

表 11.6 是在 TDT2004 层次化话题识别任务中提交的 5 个英文系统的结果，语料中一共包含 278 109 篇英文报道。从评测的结果可以看出，基于“锥形”的有向无环图结构，采取不同的参数 q 后，虽然具体的宽度和深度有所不同，但各个系统的效果差别并不大，这说明算法对于参数 q 不敏感，因而算法是比较稳定的。

表 11.6 TDT2004 中 HTD 任务所提交结果的参数与效果

提交结果	θ	q	k	$(\text{Cost(detect)})_{\text{Norm}}$	$(\text{Cost(travel)})_{\text{Norm}}$	MinimumCost
ICT3a	0.35	0.05	$+\infty$	0.0857	0.1049	0.0984
ICT3b	0.35	0.06	$+\infty$	0.1055	0.1030	0.1039
ICT3c	0.35	0.07	$+\infty$	0.1018	0.1016	0.1017
ICT3d	0.35	0.08	$+\infty$	0.0775	0.0931	0.0878
ICT3e	0.35	0.09	$+\infty$	0.1036	0.0936	0.0970

11.7 本章小结

本章研究了基于事件的信息组织技术。首先分析了宏观粒度事件的特点，并根据这些特点提出三种话题特征调整策略。主要考察了话题的时序属性、事件角色特征词对话题识别的影响、指代消解在话题识别中的应用。基于话题具有层次包含关系的特点，在话题识别的基础上，进一步研究了话题的层次化组织方法。为此提出基于多层聚类的层次化话题生成算法，该算法主要通过相似度阈值来控制生成每一粒度层次的话题节点，然后再自底向上地把各个层次的节点组织起来。用户就可按照层次结构进行浏览，实验结果表明所提出的方法是有效的。

相关工作与扩展阅读

我们关注的人物模型的同一性判别是在事先没有先验知识的情况下进行的，属于非监督式机器学习的过程。同一性判别大体就是一个聚类的过程，总的思路是观察围绕这个人物姓名的词场，通过词场的某种特征反映不同的人物实体。不过与普通的聚类场景相比，人物模型的同一性判别有其自身的特点。不同的人物对应的词场划分不尽相同，词场内部的各个特征所起的作用也不同，所以只能结合具体的场景对普通聚类算法进行调整，是聚类问题的特殊应用。人物模型片段中有各种类型的数据，有的是通过信息抽取得到的，也有的是直接转化成向量空间产生的，因此同一性判别必须有效地对这些混合类型的数据进行处理。

人物同一性判别大体是一个聚类过程。聚类分析已经被广泛地研究，它能把数据点分成若干个组，使同一组内的数据点相似度比较大，不同组内的点相似度较小。传统的聚类方法假定数据点都是数值化的特征，然而当聚类的应用从科学和

工程领域扩展到医学、商业和社会等领域时，就不得不处理属性特征，如颜色、形状、性别等。尽管这些属性之间的相似度也可以转化为数值形式，然而任何类似的转换都是困难和主观的。为此，人们进行了专门的研究。在属性聚类方面，比较典型的算法有 K-modes、STIRR、CACTUS、ROCK 等。K-modes 可以看做是 K-means 算法向属性聚类方向的扩展，它采用模式来代替均值以便处理类属性特征，并采用基于频率的方法在聚类过程中来更新模式。在 K-modes 算法基础上，Jollois 等(2002)提出了一个适应性混合模型来对属性特征进行聚类，这样就给 K-modes 算法的评价标准提供了概率解释。对于本章关注的人物追踪来说，预处理阶段生成的人物模型片段中，属性信息往往是稀疏、不充分的，还无法完全脱离普通文本的数值相似度计算，应该采取普通词场的数值相似度计算与属性特征的布尔匹配计算相结合的方法。在聚类的研究中，只有很少数算法关注混合类型数据的聚类，大多数的解决方法集中于以下三类。

① 把属性特征转化为数值特征，然后采用数值特征的相似度(距离)计算公式来计算。然而，在大多数情况下这样的转换是很难找到实际意义的。另外，转换的过程也往往带来计算维度的增加，并造成聚类结果的下降。

② 把数值特征离散化，然后应用属性聚类的方法，如 K-modes。然而在离散化过程中经常伴随着重要信息的损失，因而造成聚类精度的下降。这种离散化的损失在监督式机器学习过程中已经得到证实。

③ 概括评价函数来同时衡量数值特征和属性特征，主要的难点在于如何概括评价函数上。K-Prototypes 是这方面比较典型的算法，它是一个基于划分的方法，主要是定义了一个混合距离函数来处理混合类型的数据。

Mann 等(2003)通过定义信息抽取模板来抽取出人物的若干属性特征，然后把各种特征应用在层次凝聚式聚类中。研究者比较了全部词模型、相关词模型、属性特征模型和扩展属性特征模型。“全部词模型”是指用所有的词汇来进行人名消歧；“相关词模型”是指对词汇先根据互信息进行特征选择，只把重要的词汇用来进行人名消歧；“属性特征模型”是指采用信息抽取出来的人物属性进行消歧；“扩展属性特征模型”是指采用信息抽取的候选特征来进行消歧。研究者人工制作了一个模拟数据集，实验发现“属性特征模型”与其他模型相结合，能够产生较好的结果。

Wan 等(2005)重点考查了在搜索引擎的返回结果中人名的消歧问题。他们同样采用层次凝聚式聚类的方法，把词汇特征、命名实体特征和属性特征的相似度线性融合在一起。类似地，Al-Kamha 等(2004)同样考查搜索引擎返回结果中人名的消歧，然而他们却采用了不同的特征集合，包括人物属性集合、页面链接集合和页面相似度集合。然后用这些集合构建了一个确定度矩阵，并在矩阵内部进行语料对的合并。Bekkerman 等(2005)提出两种非监督式算法框架来发现那些指

向同一个人物的同名页面：一个基于页面的链接关系，另一个采用一种多路分布式聚类方法(MDC)。然而，他们的应用场景局限于判别一个已知的社会网络中同名人物的页面，在这个网络中人们需要有一定的联系。

另外，许荣华等(2009)针对事件抽取获得的单个元事件无法完整描述主题事件的特点，提出了一种主题事件的融合方法，通过该方法将与同一主题相关的所有元事件整合在一起，以层次化的形式表示。首先定义了一种事件融合框架 TEFF。该框架根据各类元事件在主题事件中的作用，将主题事件以层次化的形式表示，同时给出元事件和主题的相关度计算方法，通过该算法来评价元事件和主题的相关度。在事件融合框架 TEFF 的指导下，通过相关度计算，实现主题事件的融合。在以 2008 年起的金融危机为主题的实验中，取得了 F 值为 77.1%的实验结果，这表明该方法能有效地对主题事件进行融合。韩永峰等(2012)针对当前有代表性的自动摘要方法是根据文本片段进行聚类。较传统方法避免了信息冗余，但网络新闻文本中有些文本片段和主题无关，影响了聚类的效果，导致最终生成的摘要不够简洁。为此该文引入事件抽取技术，提出一种基于事件抽取的网络新闻多文档自动摘要方法。该方法首先通过二元分类器辨析出文本中的事件和非事件。然后通过聚类将文档原来以段落或句子为单位的物理划分转化为以事件为单位的内容逻辑划分，最后通过主旨事件抽取、排序及润色，生成摘要。实验结果表明，该方法是有效的。刘茂福(2010)针对事件的抽取式摘要方法一般首先抽取那些描述重要事件的句子，然后把它们重组并生成摘要，将事件定义为事件项以厦与其关联的命名实体，并聚焦从外部语义资源获取的事件项语义关系。首先基于事件项语义关系创建事件项语义关系图并使用改进的 DBSCAN 算法对事件项进行聚类，接着为每类选择一个代表事件项或者选择一类事件项来表示文档集的主题。最后从文档抽取那些包含代表项并且最重要的句子生成摘要。该文的实验结果证明在多文档自动摘要中考虑事件项语义关系是从要的和可行的。冯礼(2008)引入事件新侧面探测方法，采用自动探测方法寻找框架中未预定义的侧面。为充分利用词性、语序及词之间的关系，使用词对特征模型进行特征提取，选择基于段落的 LSA 聚类算法来实现新侧面探测。根据原型系统在突发事件语料库上的测试结果，提出的方法被证明是切实可行的，对于突发事件新闻要素的抽取达到了较高的正确率和召回率。事件新侧面探测的结果较好地表现了单个事件的特性和同类事件未包含在框架内的某些共性。实验结果证明本研究的应用前景。

参 考 文 献

冯礼．2008．基于事件框架的突发事件信息抽取．上海交通大学博士学位论文．

韩永峰，许旭阳，李弼程，等．2012．基于事件抽取的网络新闻多文档自动摘要．中文信息学报，26(1)：58-66.

刘茂福，李文捷，姬东鸿，等．2010. 基于事件项语义图聚类的多文档摘要方法．中文信息学报，24(5)：77-84.

许荣华，吴刚，李培峰，等．2009. 基于事件框架的主题事件融合研究．计算机应用研究，26(12)：4542-4545.

Al-Kamha R，Embley D W. 2004. Grouping Search Engine Returned Citations for Person Name Queries// Proceedings of WIDM'04.

Bekkerman R，McCallum A. 2005. Disambiguating web appearances of people in a social network// Proceedings of WWW.

Jollois F，Nadif M. 2002. Clustering large categorical data// Proceedings of PAKDD.

Luger G，Stubblefield W. 1997. Artificial Intelligence：Structures and Strategies for Complex Problem Solving. New York：Addison Wesley.

Mann G S，Yarowsky D. 2003. Unsupervised personal name disambiguation// Proceedings of the 7th Conference on Natural Language Learning at HLT-NAACL.

NIST. 2004. The 2004 topic detection and tracking (TDT-2004) task definition and evaluation plan// National Institute of Standards and Technology.

Wan X J，Gao J F，et al. 2005. Person resolution in person search results：WebHawk// Proceedings of the ACM 14th Conference on Information and Knowledge Management.

附录 A　ICTCLAS/NLPIR 2014 汉语分词系统介绍

A.1　NLPIR/ICTCLAS2014 分词系统简介

词法分析是自然语言处理的基础与关键。张华平博士在多年研究工作的基础上，研制出 NLPIR 分词系统，主要功能包括中文分词、英文分词、词性标注、命名实体识别、新词识别、关键词提取，同时支持用户专业词典与微博分析。NLPIR 系统支持多种编码(GBK 编码、UTF8 编码、BIG5 编码)、多种操作系统(Windows、Linux、FreeBSD 等)、多种开发语言平台(C/C++/C#、Java、Python、Hadoop 等)。

NLPIR 分词系统前身是 2000 年发布的 ICTCLAS 词法分析系统。从 2009 年开始，为了方便推广 NLPIR 自然语言处理与信息检索共享平台，调整命名为 NLPIR 分词系统。张华平博士先后倾力十余年，内核升级十余次，先后获得了 2010 年钱伟长中文信息处理科学技术奖一等奖(附图 A.1)，2003 年国际



附图 A.1　NLPIR/ICTCLAS 获得钱伟长中文信息处理科学技术奖一等奖

SIGHAN 分词大赛综合第一名，2002 年国内 973 评测综合第一名。全球用户突破 30 万，包括中国移动、华为、中搜、3721、NEC、中华商务网、硅谷动力、云南日报等企业，清华大学、新疆大学、华南理工大学、麻省理工大学等机构。同时，ICTCLAS 也被《人民日报》等多家媒体报道。

我们提供各类二次开发接口，特别欢迎相关的科研人员、工程技术人员使用，并承诺非商用应用永久免费的共享策略(http://ictclas.nlpir.org/)交流。

A.2　NLPIR/ICTCLAS2014 分词系统主要功能介绍

1) 中英文混合分词功能

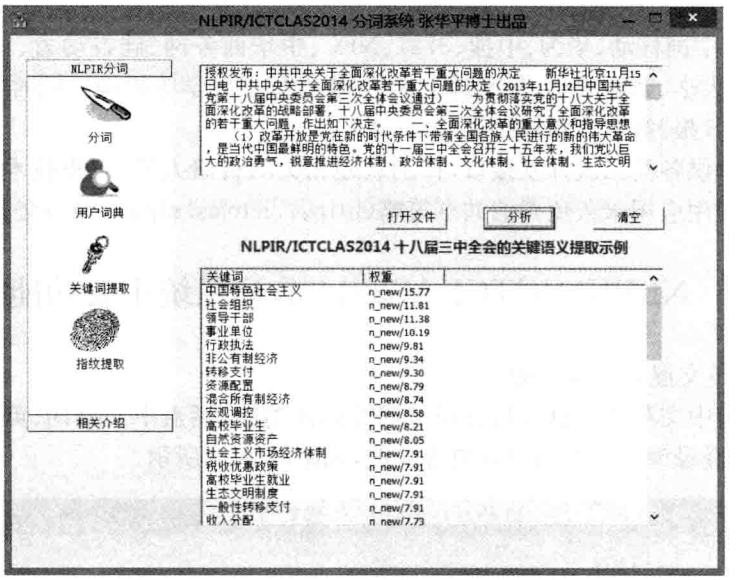
自动对中文英文信息进行分词与词性标注功能，涵盖中文分词、英文分词、词性标注、未登录词识别与用户词典等功能，如附图 A.2 所示



附图 A.2　中英文混合分词展示

2) 关键词提取功能

采用交叉信息熵的算法自动计算关键词，包括新词与已知词，下面是对十八届三中全会报告部分内容关键词的提取结果(附图 A.3)。



附图 A.3　十八届三中全会报告关键词的提取结果

3）新词识别与自适应分词功能

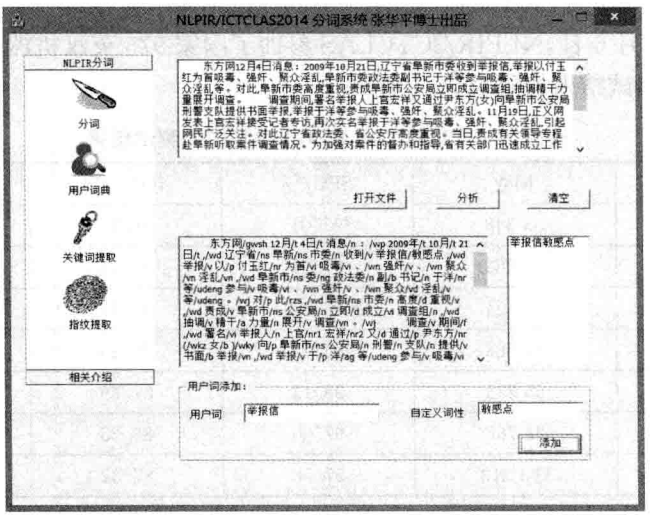
从较长的文本内容中，基于信息交叉熵自动发现新特征语言，并自适应测试语料的语言概率分布模型，实现自适应分词（附图 A.4）。



附图 A.4　自动识别新词等功能

4) 用户专业词典功能

可以单条导入用户词典，也可以批量导入用户词典。例如，可以定“举报信 敏感点”，其中举报信是用户词，敏感点是用户自定义的词性标记(附图 A.5)。



附图 A.5　判别用户定义词“举报信”，设置为自定义词性“敏感点”

5) 微博分词功能

对博主 ID 进行 nr 标示，对转发的会话进行自动分割标示(标示为 ssession)，URL 以及 Email 进行自动标引(附图 A.6)。



附图 A.6　微博分词示例

A.3 NLPIR/ICTCLAS2014 分词系统评测

(1) NLPIR/ICTCLAS 在 973 评测中的测试结果

2002 年 7 月 6 日,NLPIR/ICTCLAS 参加了国家 973 英汉机器翻译第二阶段的开放评测,测试结果如附表 A.1 所示。

附表 A.1 ICTCLAS 在 973 评测中的测试结果

领域	词数	SEG/%	TAG1/%	RTAG/%
体育	33 348	97.01	86.77	89.31
国际	59 683	97.51	88.55	90.78
文艺	20 524	96.40	87.47	90.59
法制	14 668	98.44	85.26	86.59
理论	55 225	98.12	87.29	88.91
经济	24 765	97.80	86.25	88.16
总计	208 213	97.58	87.32	89.42

① 数据来源为国家 973 英汉机器翻译第二阶段评测的评测总结报告。

② 标注相对正确率 RTAG=TAG1/SEG * 100%。

③ 我们采取的词性标注集和专家组的标注集有较大出入,所以词性标注的正确率不具可比性。

④ 专家组的开放评测结果表明,基于 HHMM 的 ICTCLAS 能解决汉语词法分析问题,和相关单位的类似系统对比,ICTCLAS 的分词结果表现出色。

(2) 第一届国际分词大赛的评测结果

为了比较和评价不同方法和系统的性能,第 41 届国际计算语言联合会(Annual Meeting of the Association for Computational Linguistics, ACL)下设的汉语特别兴趣研究组(the ACL Special Interest Group on Chinese Language Processing, SIGHAN) 于 2003 年 4 月举办了第一届国际汉语分词评测大赛。报名参赛的分别是来自中国、美国等 6 个国家和地区,共计 19 家研究机构,最终提交结果的是 12 家参赛队伍。

大赛采取大规模语料库测试,进行综合打分的方法,语料库和标准分别来自北京大学(简体版)、宾州树库(简体版)、香港城市大学(繁体版),台湾"中央研究院"(繁体版)。每家标准分两个任务(Track):受限训练任务(Close Track)和非受限训练任务(Open Track)。

NLPIR/ICTCLAS 分别参加了简体的所有四项任务和繁体的受限训练任务。其中在宾州树库受限训练任务中综合得分 0.881,名列第一;北京大学受限训练任

务中综合得分 0.951，名列第一；北京大学受限训练任务中综合得分 0.953，名列第二。值得注意的是，我们在短短的两天之内，采取 ICTCLAS 简体版的内核代码，将多层隐马尔可夫模型推广到繁体分词当中，同样取得了 0.938 的综合得分。

（3）NLPIR/ICTCLAS 的评测结果

我们利用了《人民日报》1998 年 1 月的新闻纯文本语料进行开放测试，ICTCLAS3.0 测试的精度与速度如附表 A.2 所示。

附表 A.2　ICTCLAS3.0 测试的精度与速度

	开放测试一	开放测试二	开放测试三
功能描述	分词	分词＋命名实体与新词识别	分词＋命名实体与新词识别＋词性标注
测试文件大小/Byte	4 092 478	4 092 478	4 092 478
时间/s	4.094000	6.467561	9.094001
核心数据所占内存/MB	5.5	7.2	8.9
速度/KB/s	999.63	632.77	450.02
精度/%	分词精度：96.56	分词精度：98.13	分词精度：98.13 词性标精度：94.63

说明：

① 测试机器配置：CPU P43.0G；内存 512M。

② 分词精度指的是正确切分的词数占正确结果总词数的百分比；词性标注精度指的是切分与词性标注均正确的词数占正确结果总词数的百分比。

③ 开放测试指的是测试样本不属于训练样本集合，否则称为封闭测试。封闭测试相当于考试试题都出自于学习过的书本，这种测试并没有实质意义，而往往有一些商家故意混淆视听，以封闭测试来冒充开放测试，制造准确率 99.5%的噱头。实际上，通过机械记忆小样本的封闭测试取得 100%的精度不存在任何问题。这一点特别请读者注意。

A.4　NLPIR/ICTCLAS 大事记

① 2000 年 5 月，张华平进入刘群教授领导的自然语言处理课题组，开始从事分词的研发，2000 年 8 月第一版研制成功并发布，并发表第一篇分词的论文。

② 2002 年 7 月，在 973 项目“图像、语音、自然语言理解与知识挖掘”专家组的评测中，在所有参评的系统中，评测得分最高（分词正确率高达 97.58%，参赛单位包括北京大学、清华大学等）。

③ 2003 年 1 月 7 日，获得国家版权局授予的软件著作权登记证书，编号为软著登字 005178 号。

④ 在 2003 年 4 月 22 日至 25 日，ICTCLAS 参加了第四十一届国际计算语言

联合会下设的汉语特别兴趣研究组举办的第一届国际汉语分词评测大赛，在参加的六项比赛中，获得了两项第一名、一项第二名（参赛单位来自于 6 个国家和地区的 12 个系统，包括微软、SYSTRAN、Pennsylvania 大学、Berkeley 大学、北京大学）。

⑤ 作为中国科学院计算技术研究所的 15 项免费技术成果之一，国内外约 30 万次以上的装机量。作为中文自然语言处理开放平台的自由软件，受到了广泛的欢迎和关注。我们提供的各种形式研究成果，在学术界和产业界得到了广泛的应用，其中包括麻省理工学院、清华大学、新疆大学、北京师范大学、中国科学技术情报研究所等科研机构，以及中国移动、华为、NCR、Adobe、NEC、人民网等各类企业。

⑥ 2004 年 7 月，推出 ICTCLAS 2.0。

⑦ 2005 年 12 月，推出 ICTCLAS 2.6。

⑧ 2006 年 4 月，推出 ICTCLAS 3.0，速度接近 1MB/s，精度 98.13%。

⑨ 2008 年初，推出 ICTCLAS 2008；开始按照年份作为版本序号。

⑩ 2010 年初，张华平博士调任北京理工大学，推出 ICTCLAS 2010，并将名称调整为 NLPIR。2010 年获得钱伟长中文信息处理科学技术奖一等奖。

⑪ 2012 年 11 月，推出 NLPIR/ICTCLAS 2013，增加了自适应分词、新词识别与关键词提取功能。第一次采用社交网络的形式发布内测，将库文件的名称统一改为 libNLPIR.so/NLPIR.dll。

⑫ 2013 年 12 月，推出 NLPIR/ICTCLAS 2014，第一次进行线下的分词用户交流大会。

附录 B　NLPIR 大数据搜索与挖掘共享开发平台

B.1　NLPIR 网络搜索与挖掘共享开发平台简介

NLPIR 网络搜索与挖掘共享开发平台针对互联网内容处理的需要，融合了自然语言理解、网络搜索和文本挖掘的技术，提供了用于技术二次开发的基础工具集。开发平台由多个中间件组成，各个中间件 API 可以无缝地融合到客户的各类复杂应用系统之中，可兼容 Windows、Linux、Android、Maemo5、FreeBSD 等不同操作系统平台，可以供 Java、C、C# 等各类开发语言使用。

NLPIR 是一套专门针对原始文本集进行处理和加工的软件，提供了中间件处理效果的可视化展示，也可以作为小规模数据的处理加工工具。用户可以使用该软件对自己的数据进行处理。

NLPIR 网络搜索与挖掘共享开发平台的十一种功能。

1）JZSearch 全文精准检索

支持文本、数字、日期、字符串等各种数据类型，多字段的高效搜索，支持 AND/OR/NOT 以及 NEAR 邻近等查询语法，支持维语、藏语、蒙语、阿拉伯、韩语等多种少数民族语言的检索。可以无缝地与现有文本处理系统与数据库系统融合。

2）新词发现

从文件集合中挖掘出内涵的新词语列表，可以用于用户专业词典的编撰，还可以进一步编辑标注，导入分词词典中，从而提高分词系统的准确度，并适应新的语言变化。

3）分词标注

对原始语料进行分词、自动识别人名地名机构名等未登录词、新词标注以及词性标注，并可在分析过程中，导入用户定义的词典。

4）统计分析与术语翻译

针对切分标注结果，系统可以自动地进行一元词频统计、二元词语转移概率统计（统计两个词左右连接的频次即概率）。针对常用的术语，会自动给出相应的英文解释。

5）大数据聚类及热点分析

能够从大规模数据中自动分析出热点事件，并提供事件话题的关键特征描述。同时，适用于长文本和短信、微博等短文本的热点分析。

6）大数据分类过滤

针对事先指定的规则和示例样本，系统自动从海量文档中筛选出符合需求的样本。

7）自动摘要

能够对单篇或多篇文章，自动提炼出内容的精华，方便用户快速浏览文本内容。

8）关键词提取

能够对单篇文章或文章集合，提取出若干个代表文章中心思想的词汇或短语，可用于精化阅读、语义查询和快速匹配等。

9）文档去重

能够快速准确地判断文件集合或数据库中是否存在相同或相似内容的记录，同时找出所有的重复记录。

10）HTML 正文提取

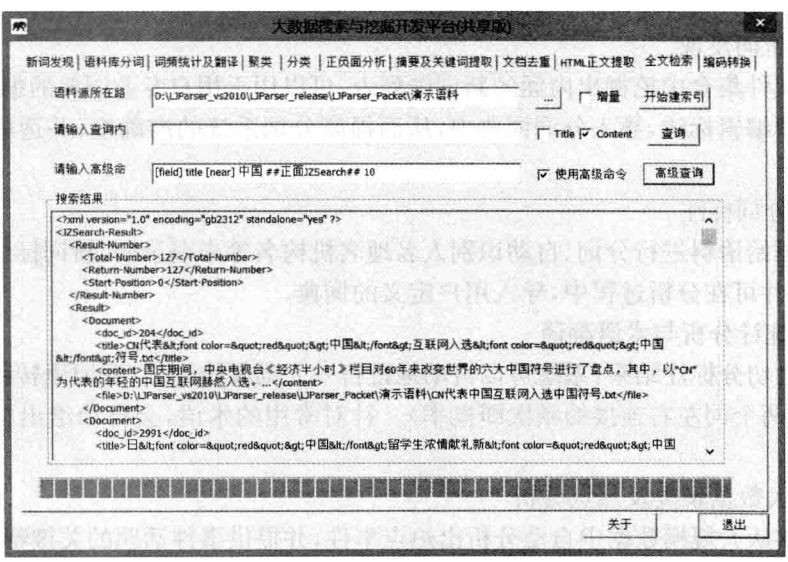
自动剔除导航性质的网页，剔除网页中的 HTML 标签和导航、广告等干扰性文字，返回有价值的正文内容。适用于大规模互联网信息的预处理和分析。

11）编码自动识别与转换

自动识别文档内容的编码，并进行自动转换，目前支持 Unicode/BIG5/UTF-8 等编码自动转换为简体的 GBK，同时将繁体 BIG5 和繁体 GBK 进行繁简转化。

B.2 开发平台工具软件操作指南

按照功能如附图 B.1 所示。



附图 B.1 NLPIR 平台之 JZSearch 大数据全文搜索功能

B. 2. 1　JZSearch 全文精准检索

选择语料文件夹，点击“开始建索引”按钮，系统对语料快速建立压缩索引；输入查询关键词，点击查询按钮，系统返回查询结果，并配以权重。

全文精准搜索的特色在于如下方面。

(1) 支持无词典索引，支持搜索维语、藏语、蒙语、阿拉伯、韩语等多种少数民族语言。

当前的搜索大部分都需要内置一个核心词库，而维语、藏语、蒙语、阿拉伯、韩语等多种少数民族语言往往缺乏相关的电子资源，整理一部词典往往费时费力。JZSearch 全文精准搜索引擎支持词典与无词典两种模式，无词典时，采用 N-Gram 模型，同样可以构建高速的索引与搜索。

(2) 支持文本、数字、日期、字符串等各种数据类型，多字段的高效搜索。

(3) 内置多种检索模型，支持多种排序策略，包括相关度、时序等。

(4) 全文索引压缩比约为 1/4，大大减少了索引的开销，提高了所有效率。

(5) 支持丰富的查询语法，支持与、或、非以及邻近运算。

支持的典型查询语法。

Sample1：[FIELD] title [AND]解放军

Sample2：[FIELD] title [AND]解放军某部发生数百人感染甲流疫情

Sample3：[FIELD] content [AND]甲型 H1N1 流感

Sample4：[FIELD] content [NEAR]张雁灵解放军

Sample5：[FIELD] content [OR]解放军甲流

Sample6：[FIELD] title [AND]解放军 [FIELD] content [NOT] 甲流

(6) 可扩展性强：支持数据库的全文搜索，以及 word、ppt、pdf、email 等各种文档格式的搜索，可以便利地构建各类网络搜索引擎服务。

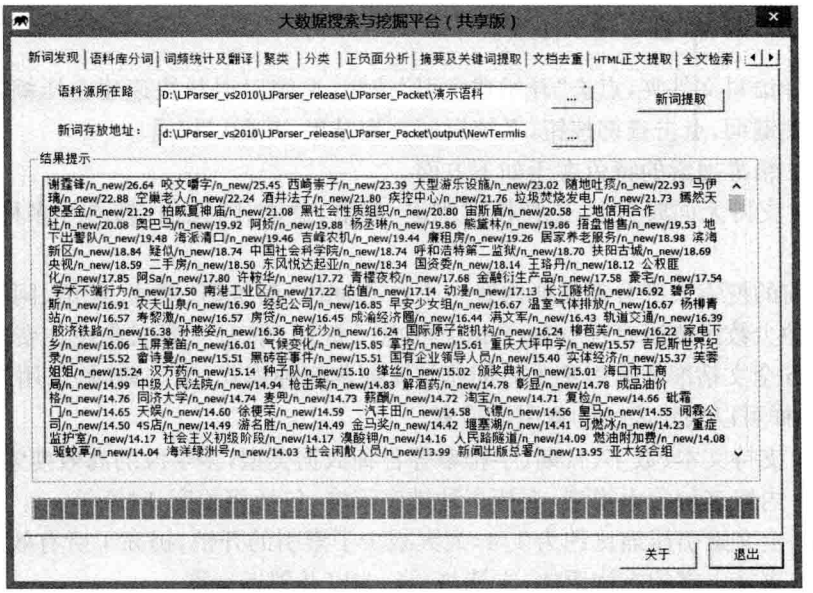
B. 2. 2　新词发现

在“语料源所在路径”输入框中输入需要提取新词的语料所在路径，语料需以 txt 格式存储在输入的语料源目录下。

如果“语料源所在路径”是通过选择文件夹方式确定，则系统会缺省指定“新词存放地址”为当前工作目录\output\NewTermlist. txt。如果“语料源所在路径”是由手动输入，则需要指定输出的“新词存放地址”。

点击“新词提取”按钮，系统开始进行发现新词的过程。结果输出到“新词存放地址”所指定的文件，另外也会输出到结果提示框中(附图 B. 2)。

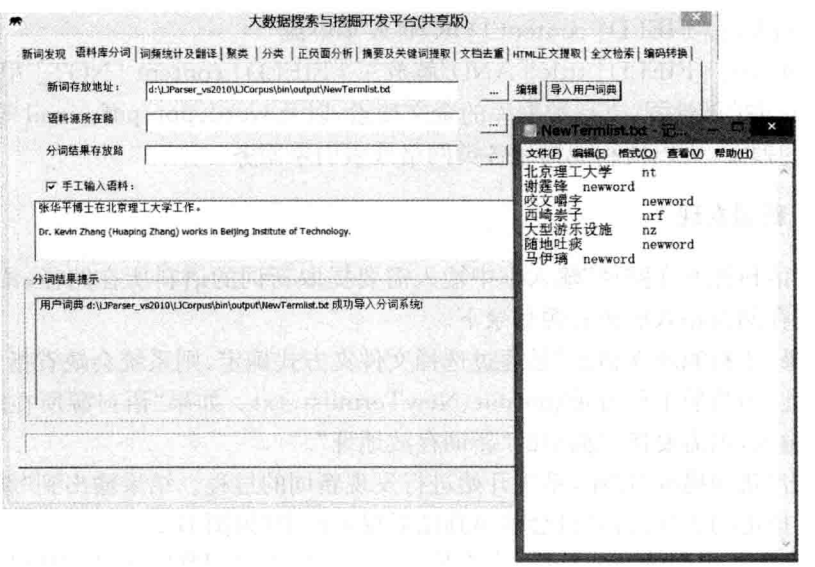
本步骤所得到的新词，可以作为分词标注器的用户词典导入，从而使分词结果更加准确。对于不需要导入新词的用户，本步骤可以跳过。



附图 B.2 NLPIR 平台之新词发现功能

B.2.3 导入用户词典，对语料进行分词及词性标注

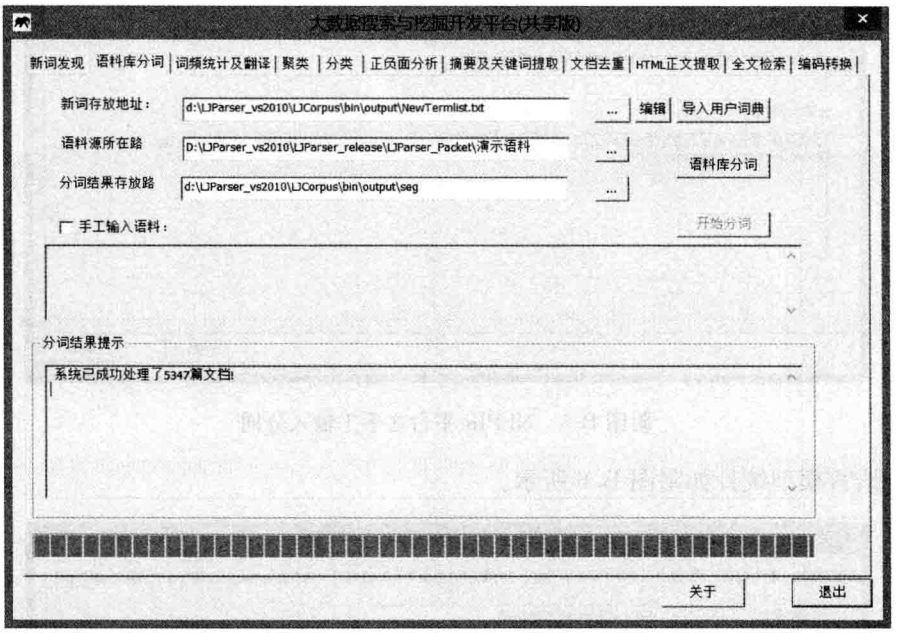
（1）导入用户词典（附图 B.3）。



附图 B.3 NLPIR 平台之用户词典导入功能

指定新词文件，用户可以对新词列表进行编辑（系统给出的标注默认为newword，用户可以根据实际情况进行校对，词性可以标注为任意字符串，系统不做限制）后，再点击“导入用户词典”，在结果提示框中会显示是否导入成功。

（2）语料库批量分词与词性标注（附图 B.4）。



附图 B.4　NLPIR 平台之语料库批量分词

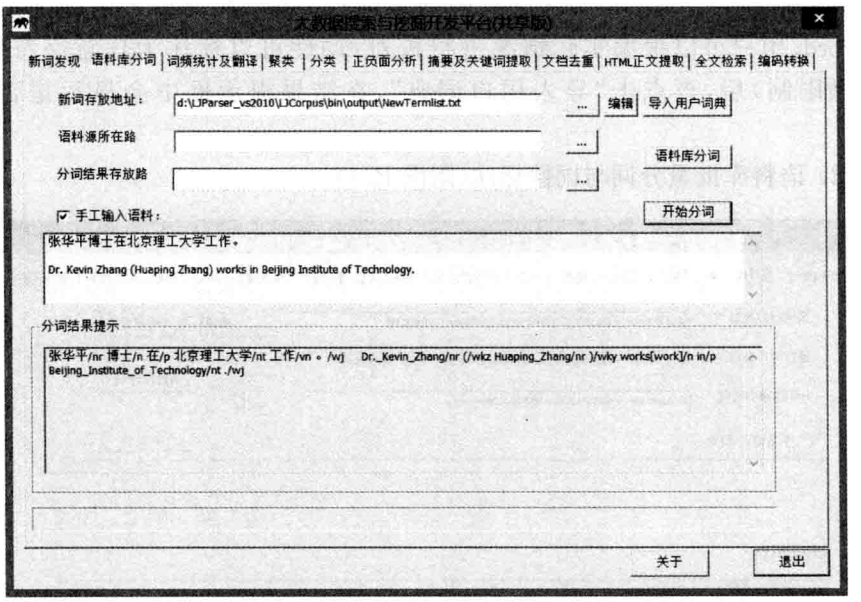
首先指定语料源所在路径，该目录下的语料可以与新词发现中所使用的语料相同，也可以不同，根据用户需求确定。

同第一步一样，选择语料源所在路径后，系统会指定默认的“分词结果存放路径”为：当前工作目录\output\seg。用户也可以指定其他输出路径。分词及词性标注结果以 txt 格式的文件存放，文件名与源语料中的文件名一致。

点击“语料库”分词，系统开始分词与词性标注。处理完成后，结果输出到“分词结果存放路径”目录下，系统会在完成时自动为用户打开该目录。

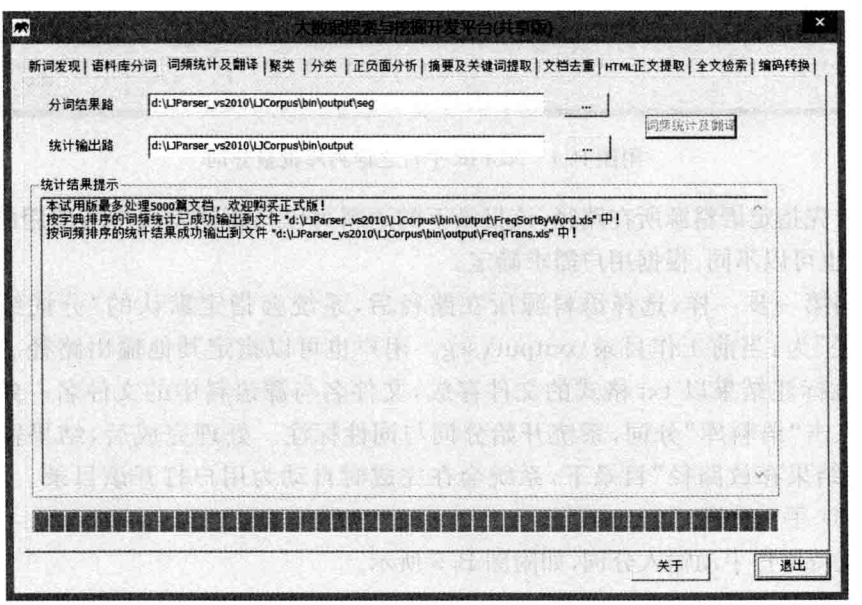
（3）手工分词。

支持用户手动输入分词，如附图 B.5 所示。



附图 B.5 NLPIR 平台之手工输入分词

语言模型统计如附图 B.6 所示。



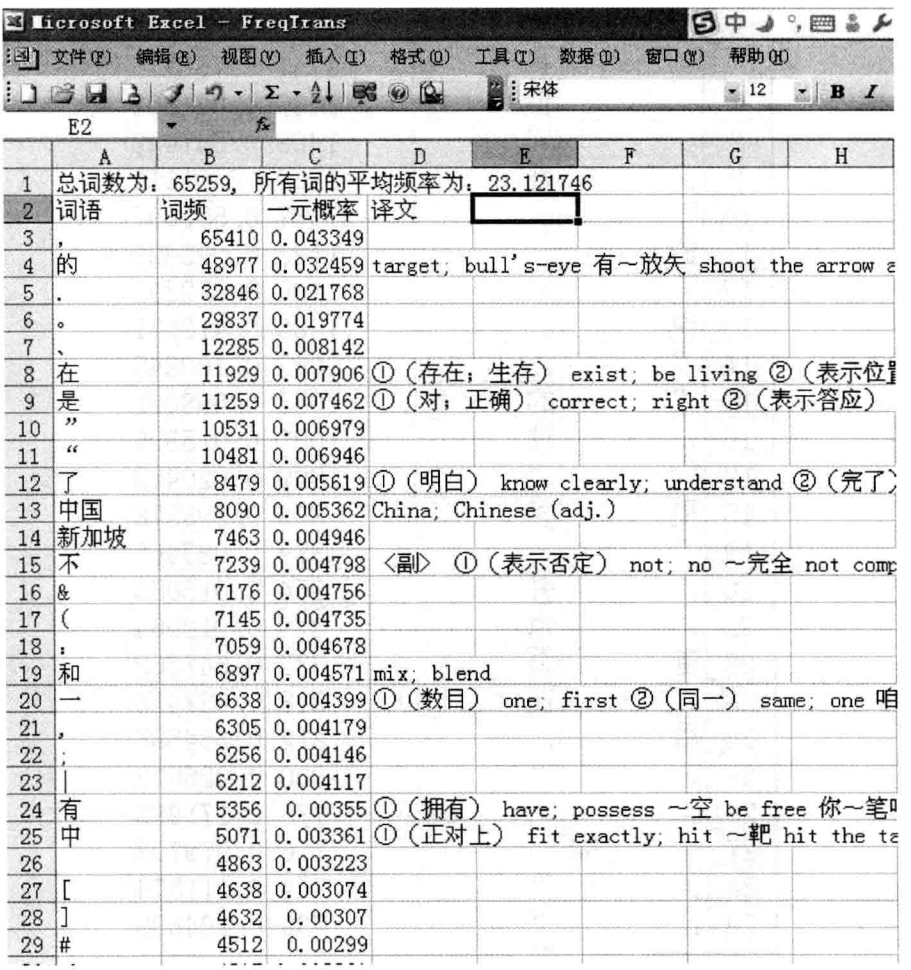
附图 B.6 NLPIR 平台之语言模型统计

B. 2. 4　词频统计及翻译

输入“分词结果路径”，该目录下的文件为第二步分词标注的结果。

同样的，指定“分词结果路径”之后，系统会指定一个默认的“统计输出路径：当前工作目录\output”。用户也可以指定其他输出路径。

点击“词频统计及翻译”按钮，系统开始统计词频、共现词对频率等信息。输出结果分别为：按照词典序排列的词频统计；按照词频大小排序的词频统计，该输出文件包含了词的英文翻译（如附图 B. 7 所示）；按照共现词对频率排列的共现词对统计文件（如附图 B. 8 所示）。



Microsoft Excel - FreqTrans

文件(F)　编辑(E)　视图(V)　插入(I)　格式(O)　工具(T)　数据(D)　窗口(W)　帮助(H)

宋体　12　B　I

E2

	A	B	C	D	E	F	G	H
1	总词数为：65259，所有词的平均频率为：23.121746							
2	词语	词频	一元概率	译文				
3	,	65410	0.043349					
4	的	48977	0.032459	target; bull's-eye 有～放矢 shoot the arrow a				
5	.	32846	0.021768					
6	。	29837	0.019774					
7	、	12285	0.008142					
8	在	11929	0.007906	①（存在；生存） exist; be living ②（表示位置				
9	是	11259	0.007462	①（对；正确） correct; right ②（表示答应）				
10	”	10531	0.006979					
11	“	10481	0.006946					
12	了	8479	0.005619	①（明白） know clearly; understand ②（完了）				
13	中国	8090	0.005362	China; Chinese (adj.)				
14	新加坡	7463	0.004946					
15	不	7239	0.004798	〈副〉 ①（表示否定） not; no ～完全 not comp				
16	&	7176	0.004756					
17	(	7145	0.004735					
18	:	7059	0.004678					
19	和	6897	0.004571	mix; blend				
20	一	6638	0.004399	①（数目） one; first ②（同一） same; one 咱				
21	,	6305	0.004179					
22	;	6256	0.004146					
23	\|	6212	0.004117					
24	有	5356	0.00355	①（拥有） have; possess ～空 be free 你～笔吗				
25	中	5071	0.003361	①（正对上） fit exactly; hit ～靶 hit the ta				
26		4863	0.003223					
27	[	4638	0.003074					
28	]	4632	0.00307					
29	#	4512	0.00299					

附图 B. 7　词频统计分析及翻译结果

Microsoft Excel - Bigrams

文件(F) 编辑(E) 视图(V) 插入(I) 格式(O)

D2 转移概率

	A	B	C	D
1	二元词对总数为：497909			
2	前一个词	后一个词	频次	转移概率
3	.	.	31848	0.969616
4	&	#	4245	0.591555
5	;	&	2204	0.352302
6	*	*	1741	0.765275
7	,	但	1433	0.021908
8	。	”	1403	0.047022
9	,	在	1250	0.01911
10	’	s	1137	0.538097
11	的	,	1123	0.022929
12	>	>	1106	0.372391
13	说	,	1083	0.412571
14	”	,	1062	0.100845
15	”	的	1033	0.098091
16	,	这	1007	0.015395
17	的	“	1000	0.020418
18	更	多	879	0.398278
19	”	。	879	0.083468
20	,	并	854	0.013056
21	,	也	848	0.012964
22	首	页	832	0.547368
23	。	在	818	0.027416
24	后	,	812	0.354585
25	。	“	800	0.026812
26	#	58	776	0.171986
27	58	;	776	0.979798
28	,	“	744	0.011374
29	:	“	739	0.104689

附图 B.8 二元词对的统计结果

B. 2. 5　大数据聚类及热点内容分析

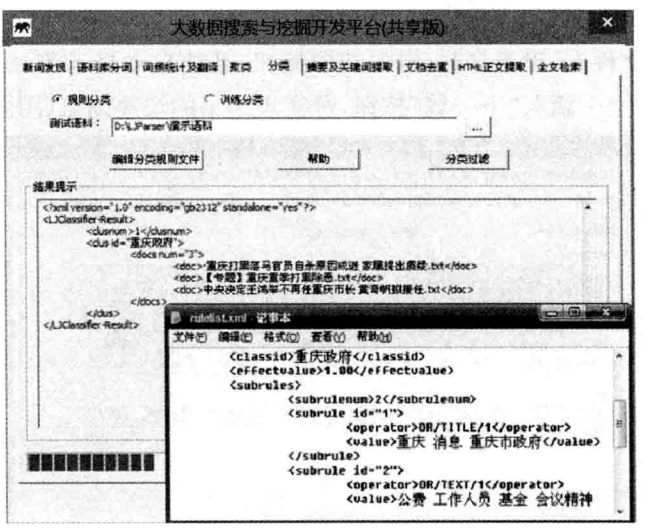
选择语料文件夹，设置参数和频繁出现的领域干扰词，点击聚类，系统返回语料所描述的热点事件话题(附图 B. 9)。



附图 B. 9　NLPIR 平台之大数据聚类功能

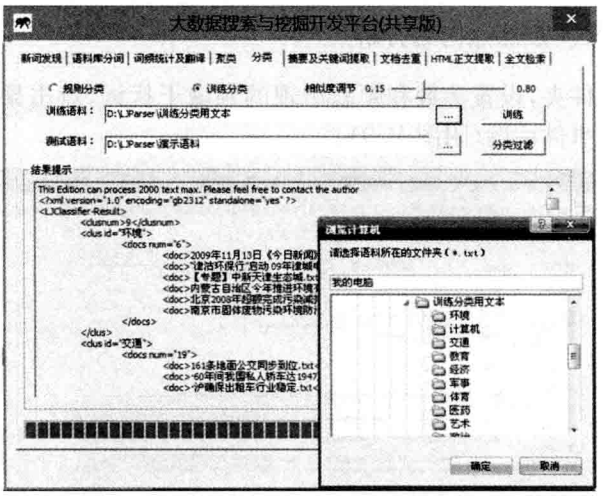
B. 2. 6　大数据过滤与分类

大数据过滤功能如附图 B. 10 所示。



附图 B. 10　NLPIR 平台之大数据过滤功能

选择语料文件夹，编辑分类规则文件(附图 B. 11 所示)，点击“分类过滤”按钮，系统返回规则过滤的结果。



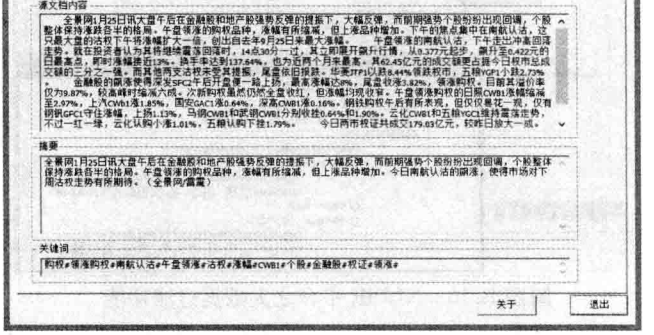
附图 B.11 NLPIR 平台之大数据自动分类

选择训练语料(各个类别需要按子文件夹排放)，点击“训练”按钮，系统进行类别特征的自学习；选择测试语料文件夹，点击“分类过滤”按钮，系统返回分类过滤的结果。

可以通过调节相似度，来控制分类过滤的内容模糊匹配程度。

B.2.7 文本摘要与关键词提取

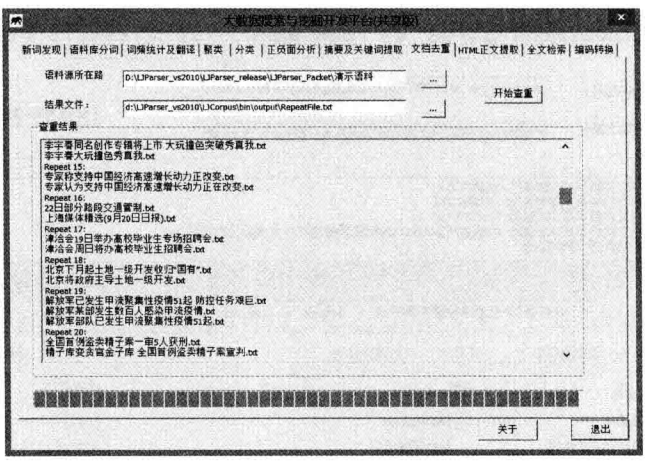
选择语料文件夹，设置参数，点击获取按钮，系统自动显示摘要和关键词的结果。通过点击“上一篇”、“下一篇”按钮，可实现结果的快速浏览(附图 B.12)。



附图 B.12 NLPIR 平台之大数据自动摘要与关键词提取功能

B. 2. 8　文档去重

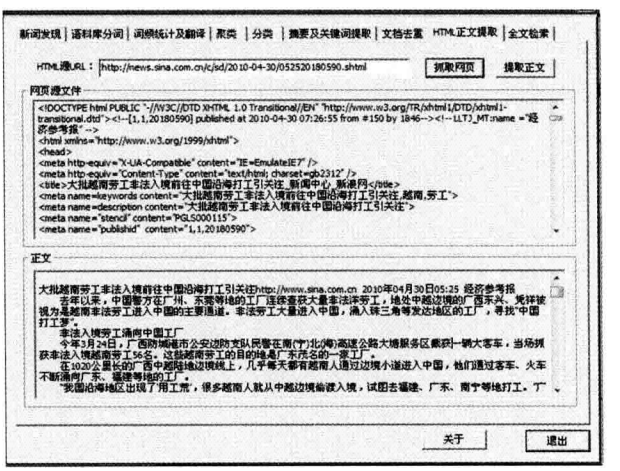
选择语料文件夹，选择结果文件存放路径，点击“开始查重”按钮，系统返回查重的结果(附图 B. 13)。



附图 B. 13　NLPIR 平台之大数据相似去重

B. 2. 9　HTML 正文解析

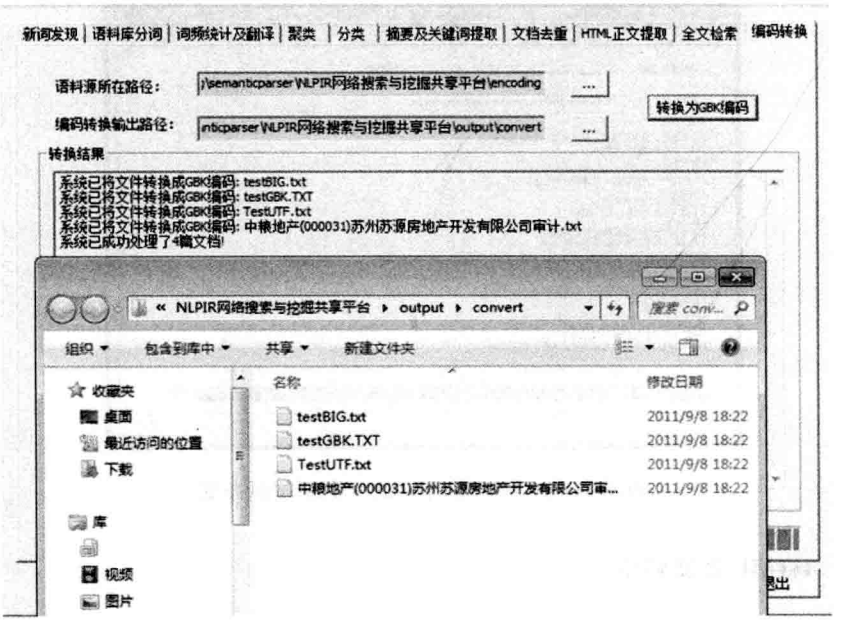
输入 URL，点击抓取按钮，下载网页源文件；然后点击提取正文按钮，系统显示正文结果，去除了大量的垃圾干扰信息(附图 B. 14)。



附图 B. 14　NLPIR 平台之 HTML 正文提取

B. 2. 10 编码识别与转换

选择待处理的文件夹，点击转换按钮，系统自动识别给定的 BIG5 文件、GBK、UTF－8 以及 Unicode 文件，最终转化为简体 GBK(附图 B. 15)。



附图 B. 15 NLPIR 平台之编码自动识别与转换